СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Теория принятия решений занимается разработкой и применением методов выбора наилучшего варианта из множества альтернатив в условиях неопределенности и ограничений. В рамках этой дисциплины особое внимание уделяется мультикритериальному анализу, где необходимо учитывать сразу несколько, порой противоречивых, критериев. Метод Паретто позволяет отобрать решения, не уступающие другим по всем параметрам. Метод Электро II применяется для ранжирования и сравнения альтернатив с использованием отношения превосходства. Метод анализа иерархий (AHP) помогает структурировать сложные задачи принятия решений, разделяя их на уровни и определяя приоритеты. Также в теории активно используются графические методы, симплексный метод линейного программирования и двойственная задача для оптимизации решений при наличии численных ограничений.

1 МЕТОД ПАРЕТО

Метод Парето, или принцип Парето, широко используется в задачах многокритериального принятия решений и оптимизации. Основная идея метода заключается в следующем:

Сравнение альтернатив по нескольким критериям:

При наличии множества вариантов (альтернатив) решения сравниваются по ряду критериев. Один вариант считается доминирующим по отношению к другому, если он по каждому критерию не хуже, а хотя бы по одному – лучше.

Парето-оптимальное множество:

Решения, которые не доминируются ни одним другим, называются Парето-оптимальными. Это означает, что нельзя улучшить хотя бы один критерий, не ухудшив при этом другой. Такое множество представляет собой набор лучших компромиссных вариантов, из которого затем может быть выбран окончательный вариант с учётом дополнительных предпочтений или ограничений.

1.1 Выбор Парето-оптимального множества

Задается множество альтернатив и их критериев со стремлениями.

Каждая альтернатива сравнивается с другими. Если найдется такая альтернатива s_j , что $s_j \succ s_i$ (то есть s_j не хуже s_i по всем критериям и хотя бы по одному критерию лучше), то s_i считается доминируемой, а s_j доминирующей.

Если для s_i ни одна другая альтернатива не доминирует над ней, s_i считается Парето-оптимальной.

Собираем все альтернативы, которые не доминируются ни одной другой, в множество, которое является Парето-оптимальным множеством.

Предметная область: Выбор космического корабля.

Таблица 1.1.1 – Альтернативы со стремлениями

	Космический корабль	**Цена (кредиты) (-)**	**Скорость (км/ч) (+)**	**Время входа в гиперпространство (сек) (-)**	**Количество орудий (шт) (+)**	**Мощность щитов (Вт) (+)**
1	TIE Fighter	20000	5000	3.0	2	100
2	TZ-24	22000	4900	3.2	4	120
3	S-100	21000	4800	3.1	3	150
4	F-T2	30000	5100	4.0	3	110
5	CR90	25000	4600	3.5	2	130
6	IL-5	26000	4700	3.7	2	100
7	FT-6	35000	4400	4.5	2	100
8	FT-8	34000	4500	4.3	3	115
9	S-13	33000	4600	4.1	2	105
10	S-SC4	32000	4700	3.9	3	125

Найдём оптимальное множество:

Таблица 1.1.2 – Нахождение оптимального множества

	1	2	3	4	5	6	7	8	9	10
1										
2										
3										
4										
5			3							
6	1	2	3							
7		2	3	4	5					
8		2	3							
9		2	3	4	5					
10			3							

Парето-оптимальное множество альтернатив {TIE Fighter, TZ-24, S-100, F-T2, CR90}.

1.2 Указание верхних/нижних границ критериев.

Парето-оптимальное множество альтернатив {TIE Fighter, TZ-24, S-100, F-T2, CR90}.

Для вариантов решений зададим границы. Цена корабля не должна превышать 22000 кредитов, скорость не менее 4800 км/ч, время входа в гиперпространство не менее 3.0 секунд, количество орудий 3 и более, мощность щитов больше 100 ватт.

Таблица 1.2 – Парето-оптимальные варианты с указанием границ

	Космический корабль	**Цена (кредиты) (-)**	**Скорость (км/ч) (+)**	**Время входа в гиперпространство (сек) (-)**	**Количество орудий (шт) (+)**	**Мощность щитов (Вт) (+)**
1	TZ-24	22000	4900	3.2	4	120
2	S-100	21000	4800	3.1	3	150

1.3 Субоптимизация

Парето-оптимальное множество альтернатив {TIE Fighter, TZ-24, S-100, F-T2, CR90}.

Главный критерий: Скорость.

Ограничения: Цена не более 25000 кредитов, время входа в гиперпространство больше 3.2 секунда, количество орудий больше 2, мощность щитов не менее 120 ватт.

Таблица 3 – Парето-оптимальные варианты

	Космический корабль	**Цена (кредиты) (-)**	**Скорость (км/ч) (+)**	**Время входа в гиперпространство (сек) (-)**	**Количество орудий (шт) (+)**	**Мощность щитов (Вт) (+)**
1	TIE Fighter	20000	5000	3.0	2	100
2	TZ-24	22000	4900	3.2	4	120
3	CR90	25000	4600	3.5	2	130
4	S-100	21000	4800	3.1	3	150
5	F-T2	30000	5100	4.0	3	110

Самый лучший корабль: TZ-24

1.4 Лексикографическая оптимизация

Упорядочим критерии по приоритету: важнейший критерий - Скорость, следующие: количество орудий, мощность щитов, цена, время входа в гиперпространство.

Таблица 4 – Альтернативы со стремлениями

	Космический корабль	**Цена (кредиты) (-)**	**Скорость (км/ч) (+)**	**Время входа в гиперпространство (сек) (-)**	**Количество орудий (шт) (+)**	**Мощность щитов (Вт) (+)**
1	TIE Fighter	20000	5000	3.0	2	100
2	TZ-24	22000	4900	3.2	4	120
3	S-100	21000	4800	3.1	3	150
4	F-T2	30000	5100	4.0	3	110
5	CR90	25000	4600	3.5	2	130
6	IL-5	26000	4700	3.7	2	100
7	FT-6	35000	4400	4.5	2	100
8	FT-8	34000	4500	4.3	3	115
9	S-13	33000	4600	4.1	2	105
10	S-SC4	32000	4700	3.9	3	125

Самый лучший корабль: F-T2.

1.5 Результаты работы программы

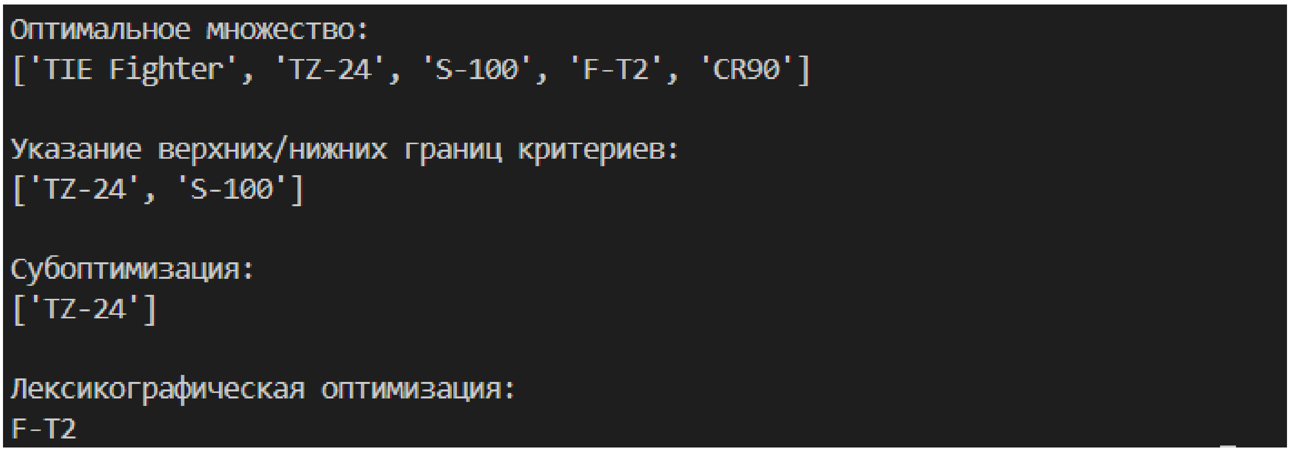

Рисунок 1.1 – Лексикографическая оптимизация.

1.6 Вывод по методу Парето

В данной работе был изучен алгоритм Парето, выполнен ручной расчёт по этому алгоритму, написана программа на языке Python, которая реализует данный алгоритм, а также изучены улучшения этого алгоритма указания верхних и нижних границ критериев, субоптимизация, лексикографическая оптимизация.

2 МЕТОД ЭЛЕКТРА II

Данный метод сравнивает все альтернативы попарно и составляет таблицу предпочтений в которую записываются коэффициенты, полученные во время сравнения. Рассматриваем все пары проектов i и j. Если по какому-либо критерию i-ый проект лучше, чем j-ый, то соответствующий критерию вес прибавляется к Pij (эти баллы символизируют выбор «За»), в противном случае — к Nij (эти баллы символизируют выбор «Против»). Затем, когда по паре i и j рассмотрены все критерии, находятся отношения Dij = Pij/Nij и Dji = Pji/Nji. Эти отношения и записываются в таблицу предпочтений.

Данный метод используется в логистике, для выбора оптимального маршрута, финансовых операциях, оценка выгодных инвестиций и в управлении множеством других проектов.

2.1 Выбор лучшего варианта

Составлена таблица критериев, по которым оцениваются проекты (Таблица 2.1).

Таблица 2.1 – Таблица критериев для оценки альтернатив

Критерии	Вес критерия	Шкала	Код	Стремление
Цена	5	31000 кредитов и более 26000 – 30000 кредитов 25000 кредитов и менее	30 20 10	min
Скорость	5	5000 км/ч и более 4900 км/ч 4800 км/ч 4700 км/ч 4600 км/ч 4500 км/ч и менее	30 25 20 15 10 5	max
Время входа в гиперпространство	4	4.1 – 4.5 сек 3.6 – 4.0 сек 3.0 – 3.5 сек	30 20 10	min
Количество орудий	4	4 шт 3 шт 2 шт	30 20 10	max
Мощность щитов	3	141 – 150 Вт 131 – 140 Вт 121 – 130 Вт 111 – 120 Вт 100 – 110 Вт	25 20 15 10 5	max

Составлена таблица оценок выбора лучшего космического коробля. Для 10-ти альтернатив заполнена Таблицу 2.2.

Таблица 2.2 – Таблица оценок по критериям

№	Варианты решений	Критерии				
		Цена	**Скорость**	**Время входа в гиперпространство**	**Количество орудий**	**Мощность щитов**
1	TIE Fighter	10	30	10	10	5
2	TZ-24	10	25	10	30	10
3	S-100	10	20	10	20	25
4	F-T2	20	30	20	20	5
5	CR90	10	10	10	10	15
6	IL-5	10	15	20	10	5
7	FT-6	30	5	30	10	5
8	FT-8	30	5	30	20	10
9	S-13	30	10	30	10	5
10	S-SC4	30	15	20	20	15
Вес		5	5	4	4	3
Стремление		min	max	min	max	max

2.2 Веса предпочтений

Рассмотрим альтернативы 1 и 2 (i=1,j=2):

P12 = 0 + 30 + 0 + 0 + 0 = 30;

N12 = 0 + 0 + 0 + 30+ 10 = 40;

D12 = P12/N12 = 30/40 < 1 – отбрасываем.

P21 = 0 + 0 + 0 + 30 + 10 = 40;

N21 = 0 + 30 + 0 + 0 + 0 = 25;

D21 = P21/N21 = 40/30 > 1 – принимаем.

Рассмотрим альтернативы 1 и 3 (i=1,j=3):

P13 = 0 + 30 + 0 + 0 + 0= 30;

N13 = 0 + 0 + 0 + 20 + 25 = 45;

D13 = P13/N13 = 30/45 < 1 – отбрасываем.

P31 = 0 + 0 + 0 + 20 + 25 = 45;

N31 = 0 + 30 + 0 + 0 + 0 = 30;

D31 = P31/N31 = 45/30 > 1 – принимаем.

Рассмотрим альтернативы 1 и 4 (i=1,j=4):

P14 = 10 + 0 + 10 + 0 + 0 = 20;

N14 = 0 + 0 + 0 + 20 + 0 = 20;

D14 = P14/N14 = 30/20 == 1 – отбрасываем.

P41 = 0 + 0 + 0 + 20 + 0 = 20;

N41 = 10 + 0 + 10 + 0 + 0 = 20;

D41 =P41/N41 = 20/30 == 1 – отбрасываем.

Рассмотрим альтернативы 1 и 5 (i=1,j=5):

P15 = 0 + 30 + 0 + 0 + 0 = 30;

N15 = 0 + 0 + 0 + 0 + 15 = 15;

D15 = P15/N15 = 30/15 > 1 – принимаем.

P51 = 0 + 0 + 0 + 0 + 15 = 15;

N51 = 0 + 30 + 0 + 0 + 0 = 30;

D51 = P51/N51 = 15/30 < 1 – отбрасываем.

Рассмотрим альтернативы 1 и 6 (i=1,j=6):

P16 = 0 + 30 + 10 + 0 + 0 = 40;

N16 = 0 + 0 + 0 + 0 + 0 = 0;

D16 = P16/N16 = 40/0 > 1 – принимаем.

P61 = 0 + 0 + 0 + 0 + 0 = 0;

N61 = 0 + 30 + 10 + 0 + 0 = 40;

D61 = P61/N61 = 0/40 < 1 – отбрасываем.

Рассмотрим альтернативы 1 и 7 (i=1,j=7):

P17 = 10 + 30 + 10 + 0 + 0 = 50;

N17 = 0 + 0 + 0 + 0 + 0 = 0;

D17 = P17/N17 = 50/0 > 1 – принимаем.

P71 = 0 + 0 + 0 + 0 + 0 = 0;

N71 = 10 + 30 + 10 + 0 + 0 =5 0;

D71 = P71/N71 = 0/50 < 1 – отбрасываем.

Рассмотрим альтернативы 1 и 8 (i=1,j=8):

P18 = 10 + 30 + 10 + 0 + 0 = 50;

N18 = 0 + 0 + 0 + 20 + 10 = 30;

D18 = P18/N18 = 50/30 > 1 – принимаем.

P81 = 0 + 0 + 0 + 20 + 10 = 30;

N81 = 10 + 30 + 10 + 0 + 0 = 50;

D81 = P81/N81 = 30/50 < 1 – отбрасываем.

Рассмотрим альтернативы 1 и 9 (i=1,j=9):

P19 = 10 + 30 + 10 + 0 + 0 = 50;

N19 = 0 + 0 + 0 + 0 + 0 = 0;

D19 = P19/N19 = 50/0 > 1 – принимаем.

P91 = 0 + 0 + 0 + 0 + 0 = 0;

N91 = 10 + 30 + 10 + 0 + 0 = 50;

D91 = P91/N91 = 0/50 < 1 – отбрасываем.

Рассмотрим альтернативы 1 и 10 (i=1,j=10):

P110 = 10 + 30 + 10 + 0 + 0 = 50;

N110 = 0 + 0 + 0 + 20 + 15 = 35;

D110 = P110/N110 = 50/35 > 1 – принимаем.

P101 = 0 + 0 + 0 + 20 + 15 = 35;

N101 = 10 + 30 + 10 + 0 + 0 = 50;

D101 = P101/N101 = 35/50 < 1 – отбрасываем.

Рассмотрим альтернативы 2 и 3 (i=2,j=3):

P23 = 0 + 25 + 0 + 30 + 0 = 55;

N23 = 0 + 0 + 0 + 0 + 25 = 25;

D23 = P23/N23 = 55/25 > 1 – принимаем.

P32 = 0 + 0 + 0 + 0 + 25 = 25;

N32 = 0 + 25 + 0 + 30 + 0 = 55;

D32 = P32/N32 = 25/55 < 1 – отбрасываем.

Рассмотрим альтернативы 2 и 4 (i=2,j=4):

P24 = 10 + 0 + 10 + 30 + 10 = 60;

N24 = 0 + 30 + 0 + 0 + 0 = 30;

D24 = P24/N24 = 60/30 > 1 – принимаем.

P42 = 0 + 30 + 0 + 0 + 0 = 30;

N42 = 10 + 0 + 10 + 30 + 10 = 60;

D42 = P42/N42 = 30/60 < 1 – отбрасываем.

Рассмотрим альтернативы 2 и 5 (i=2,j=5):

P25 = 0 + 25 + 0 + 30 + 0 = 55;

N25 = 0 + 0 + 0 + 0 + 15 = 15;

D25 = P25/N25 = 55/15 > 1 – принимаем.

P52 = 0 + 0 + 0 + 0 + 15 = 15;

N52 = 0 + 25 + 0 + 30 + 0 = 55;

D52 = P52/N52 = 15/55 < 1 – отбрасываем.

Рассмотрим альтернативы 2 и 6 (i=2,j=6):

P26 = 0 + 25 + 10 + 30 + 10 = 65;

N26 = 0 + 0 + 0 + 0 + 0 = 0;

D26 = P26/N26 = 65/0 > 1 – принимаем.

P62 = 0 + 0 + 0 + 0 + 0 = 0;

N62 = 0 + 25 + 10 + 30 + 10 = 65;

D62 = P62/N62 = 0/65 < 1 – отбрасываем.

Рассмотрим альтернативы 2 и 7 (i=2,j=7):

P27 = 10 + 25 + 10 + 30 + 10 = 85;

N27 = 0 + 0 + 0 + 0 + 0 = 0;

D27 = P27/N27 = 85/0 > 1 – принимаем.

P72 = 0 + 0 + 0 + 0 + 0 = 0;

N72 = 10 + 25 + 10 + 30 + 10 = 85;

D72 = P72/N72 = 0/85 < 1 – отбрасываем.

Рассмотрим альтернативы 2 и 8 (i=2,j=8):

P28 = 10 + 25 + 10 + 30 + 0 = 75;

N28 = 0 + 0 + 0 + 0 + 0 = 0;

D28 = P28/N28 = 75/0 > 1 – принимаем.

P82 = 0 + 0 + 0 + 0 + 0 = 0;

N82 = 10 + 25 + 10 + 30 + 0 = 75;

D82 = P82/N82 = 0/75 < 1 – отбрасываем.

Рассмотрим альтернативы 2 и 9 (i=2,j=9):

P29 = 10 + 25 + 10 + 30 + 10 = 85;

N29 = 0 + 0 + 0 + 0 + 0 = 0;

D29 = P29/N29 = 85/0 > 1 – принимаем.

P92 = 0 + 0 + 0 + 0 + 0 = 0;

N92 = 10 + 25 + 10 + 30 + 10 = 85;

D92 = P92/N92 = 0/85 < 1 – отбрасываем.

Рассмотрим альтернативы 2 и 10 (i=2,j=10):

P210 = 10 + 25 + 10 + 30 + 0 = 75;

N210 = 0 + 0 + 0 + 0 + 15 = 15;

D210 = P210/N210 = 75/15 > 1 – принимаем.

P102 = 0 + 0 + 0 + 0 + 15 = 15;

N102 = 10 + 25 + 10 + 30 + 0 = 75;

D102 = P102/N102 = 15/75 < 1 – отбрасываем.

Рассмотрим альтернативы 3 и 4 (i=3,j=4):

P34 = 10 + 0 + 10 + 0 + 25 = 45;

N34 = 0 + 30 + 0 + 0 + 0 = 30;

D34 = P34/N34 = 45/30 > 1 – принимаем.

P43 = 0 + 30 + 0 + 0 + 0 = 30;

N43 = 10 + 0 + 10 + 0 + 25 = 45;

D43 = P43/N43 = 30/45 < 1 – отбрасываем.

Рассмотрим альтернативы 3 и 5 (i=3,j=5):

P35 = 0 + 20 + 0 + 20 + 25 = 65;

N35 = 0 + 0 + 0 + 0 + 0 = 0;

D35 = P35/N35 = 65/0 > 1 – принимаем.

P53 = 0 + 0 + 0 + 0 + 0 = 0;

N53 = 0 + 20 + 0 + 20 + 25 = 65;

D53 = P53/N53 = 0/65 < 1 – отбрасываем.

Рассмотрим альтернативы 3 и 6 (i=3,j=6):

P36 = 0 + 20 + 10 + 20 + 25 = 75;

N36 = 0 + 0 + 0 + 0 + 0 = 0;

D36 = P36/N36 = 75/0 > 1 – принимаем.

P63 = 0 + 0 + 0 + 0 + 0 = 0;

N63 = 0 + 20 + 10 + 20 + 25 = 75;

D63 = P63/N63 = 0/75 < 1 – отбрасываем.

Рассмотрим альтернативы 3 и 7 (i=3,j=7):

P37 = 10 + 20+ 10 + 20 + 25 = 85;

N37 = 0 + 0 + 0 + 0 + 0 = 0;

D37 = P37/N37 = 85/0 > 1 – принимаем.

P73 = 0 + 0 + 0 + 0 + 0 = 0;

N73 = 10 + 20+ 10 + 20 + 25 = 85;

D73 = P73/N73 = 0/85 < 1 – отбрасываем.

Рассмотрим альтернативы 3 и 8 (i=3,j=8):

P38 = 10 + 20 + 10 + 0 + 25 = 65;

N38 = 0 + 0 + 0 + 0 + 0 = 0;

D38 = P38/N38 = 65/0 > 1 – принимаем.

P83 = 0 + 0 + 0 + 0 + 0 = 0;

N83 = 10 + 20 + 10 + 0 + 25 = 65;

D83 = P83/N83 = 0/65 < 1 – отбрасываем.

Рассмотрим альтернативы 3 и 9 (i=3,j=9):

P39 = 10 + 20 + 10 + 20 + 25 = 85;

N39 = 0 + 0 + 0 + 0 + 0 = 0;

D39 = P39/N39 = 85/0 > 1 – принимаем.

P93 = 0 + 0 + 0 + 0 + 0 = 0;

N93 = 10 + 20 + 10 + 20 + 25 = 85;

D93 = P93/N93 = 0/85 < 1 – отбрасываем.

Рассмотрим альтернативы 3 и 10 (i=3,j=10):

P310 = 10 + 20 + 10 + 0 + 25 = 65;

N310 = 0 + 0 + 0 + 0 + 0 = 0;

D310 = P310/N310 = 65/0 > 1 – принимаем.

P103 = 0 + 0 + 0 + 0 + 0 = 0;

N103 = 10 + 20 + 10 + 0 + 25 = 65;

D103 = P103/N103 = 0/65 < 1 – отбрасываем.

Рассмотрим альтернативы 4 и 5 (i=4,j=5):

P45 = 0 + 30 + 0 + 20 + 0 = 50;

N45 = 10 + 0 + 10 + 0 + 15 = 35;

D45 = P45/N45 = 50/35 > 1 – принимаем.

P54 = 10 + 0 + 10 + 0 + 15 = 35;

N54 = 0 + 30 + 0 + 20 + 0 = 50;

D54 = P54/N54 = 35/50 < 1 – отбрасываем.

Рассмотрим альтернативы 4 и 6 (i=4,j=6):

P46 = 0 + 30 + 0 + 20 + 0 = 50;

N46 = 10 + 0 + 0 + 0 + 0 = 10;

D46 = P46/N46 = 50/10 > 1 – принимаем.

P64 = 10 + 0 + 0 + 0 + 0 = 10;

N64 = 0 + 30 + 0 + 20 + 0 = 50;

D64 = P64/N64 = 10/50 < 1 – отбрасываем.

Рассмотрим альтернативы 4 и 7 (i=4,j=7):

P47 = 20 + 30 + 20 + 20 + 0 = 90;

N47 = 0 + 0 + 0 + 0 + 0 = 0;

D47 = P47/N47 = 90/0 > 1 – принимаем.

P74 = 0 + 0 + 0 + 0 + 0 = 0;

N74 = 20 + 30 + 20 + 20 + 0 = 90;

D74 = P74/N74 = 0/90 < 1 – отбрасываем.

Рассмотрим альтернативы 4 и 8 (i=4,j=8):

P48 = 20 + 30 + 20 + 0 + 0 = 70;

N48 = 0 + 0 + 0 + 0 + 10 = 10;

D48 = P48/N48 = 70/10 > 1 – принимаем.

P84 = 0 + 0 + 0 + 0 + 10 = 10;

N84 = 20 + 30 + 20 + 0 + 0 = 70;

D84 = P84/N84 = 10/70 < 1 – отбрасываем.

Рассмотрим альтернативы 4 и 9 (i=4,j=9):

P49 = 20 + 30 + 20 + 20 + 0 = 90;

N49 = 0 + 0 + 0 + 0 + 0= 0;

D49 = P49/N49 = 90/0 > 1 – принимаем.

P94 = 0 + 0 + 0 + 0 + 0= 0;

N94 = 20 + 30 + 20 + 20 + 0 = 90;

D94 = P94/N94 = 0/90 < 1 – отбрасываем.

Рассмотрим альтернативы 4 и 10 (i=4,j=10):

P410 = 20 + 30 + 0 + 0 + 0 = 50;

N410 = 0 + 0 + 0 + 0 + 15 = 15;

D410 = P410/N410 = 50/15 > 1 – принимаем.

P104 = 0 + 0 + 0 + 0 + 15 = 15;

N104 = 20 + 30 + 0 + 0 + 0 = 60;

D104 = P104/N104 = 15/50 < 1 – отбрасываем.

Рассмотрим альтернативы 5 и 6 (i=5,j=6):

P56 = 0 + 0 + 10 + 0 + 15 = 25;

N56 = 0 + 15 + 0 + 0 + 0 = 15;

D56 = P56/N56 = 25/15 > 1 – принимаем.

P65 = 0 + 15 + 0 + 0 + 0 = 15;

N65 = 0 + 0 + 10 + 0 + 15 = 25;

D65 = P65/N65 = 15/25 < 1 – отбрасываем.

Рассмотрим альтернативы 5 и 7 (i=5,j=7):

P57 = 10 + 10 + 10 + 0 + 15 = 35;

N57 = 0 + 0 + 0 + 0 + 0 = 0;

D57 = P57/N57 = 35/0 > 1 – принимаем.

P75 = 0 + 0 + 0 + 0 + 0 = 0;

N75 = 10 + 10 +1 0 + 0 + 15 = 35;

D75 = P75/N75 = 0/35 < 1 – отбрасываем.

Рассмотрим альтернативы 5 и 8 (i=5,j=8):

P58 = 10 + 10 + 10 + 0 + 15 = 45;

N58 = 0 + 0 + 0 + 20 + 0 = 20;

D58 = P58/N58 = 35/20 > 1 – принимаем.

P85 = 0 + 0 + 0 + 20 + 0 = 20;

N85 = 10 + 10 + 10 + 0 + 15 = 45;

D85 = P85/N85 = 20/35 < 1 – отбрасываем.

Рассмотрим альтернативы 5 и 9 (i=5,j=9):

P59 = 10 + 0 + 10 + 0 + 15 = 35;

N59 = 0 + 0 + 0 + 0 + 0 = 0;

D59 = P59/N59 = 35/0 > 1 – принимаем.

P95 = 0 + 0 + 0 + 0 + 0 = 0;

N95 = 10 + 0 + 10 + 0 + 15 = 35;

D95 = P95/N95 = 0/35 < 1 – отбрасываем.

Рассмотрим альтернативы 5 и 10 (i=5,j=10):

P510 = 10 + 0 + 10 + 0 + 0 = 20;

N510 = 0 + 15 + 0 + 20 + 0 = 35;

D510 = P510/N510 = 20/35 < 1 – отбрасываем.

P105 = 0 + 15 + 0 + 20 + 0 = 35;

N105 = 10 + 0 + 10 + 0 + 0 = 20;

D105 = P105/N105 = 35/20 > 1 – принимаем.

Рассмотрим альтернативы 6 и 7 (i=6,j=7):

P67 = 10 + 15 + 20 + 0 + 0 = 45;

N67 = 0 + 0 + 0 + 0 + 0 = 0;

D67 = P67/N67 = 45/0 > 1 – принимаем.

P76 = 0 + 0 + 0 + 0 + 0 = 0;

N76 = 10 + 15 + 20 + 0 + 0 = 45;

D76 = P76/N76 = 0/45 < 1 – отбрасываем.

Рассмотрим альтернативы 6 и 8 (i=6,j=8):

P68 = 10 + 15 + 20 + 0 + 0 = 45;

N68 = 0 + 0 + 0 + 20 + 10 = 30;

D68 = P68/N68 = 45/30 > 1 – принимаем.

P86 = 0 + 0 + 0 + 20 + 10 = 30;

N86 = 10 + 15 + 20 + 0 + 0 = 45;

D86 = P86/N86 = 30/45 < 1 – отбрасываем.

Рассмотрим альтернативы 6 и 9 (i=6,j=9):

P69 = 10 + 15 + 20 + 0 + 0 = 45;

N69 = 0 + 0 + 0 + 0 + 0 = 0;

D69 = P69/N69 = 45/0 > 1 – принимаем.

P96 = 0 + 0 + 0 + 0 + 0 = 0;

N96 = 10 + 15 + 20 + 0 + 0 = 45;

D96 = P96/N96 = 0/45 < 1 – отбрасываем.

Рассмотрим альтернативы 6 и 10 (i=6,j=10):

P610 = 10 + 0 + 0 + 0 + 0 = 10;

N610 = 0 + 0 + 0 + 20 + 15 = 35;

D610 = P610/N610 = 10/35 < 1 – отбрасываем.

P106 = 0 + 0 + 0 + 20 + 15 = 35;

N106 = 10 + 0 + 0 + 0 + 0 = 10;

D106 = P106/N106 = 35/10 > 1 – принимаем.

Рассмотрим альтернативы 7 и 8 (i=7,j=8):

P78 = 0 + 0 + 0 + 0 + 0 = 0;

N78 = 0 + 0 + 0 + 20 + 10 = 30;

D78 = P78/N78 = 0/30 < 1 – отбрасываем.

P87 = 0 + 0 + 0 + 20 + 10 = 30;

N87 = 0 + 0 + 0 + 0 + 0 = 0;

D87 = P87/N87 = 30/0 > 1 – принимаем.

Рассмотрим альтернативы 7 и 9 (i=7,j=9):

P79 = 0 + 0 + 0 + 0 + 0 = 0;

N79 = 0 + 10 + 0 + 0 + 0 = 0;

D79 = P79/N79 = 0/10 < 1 – отбрасываем.

P97 = 0 + 10 + 0 + 0 + 0 = 0;

N97 = 0 + 0 + 0 + 0 + 0 = 0;

D97 = P97/N97 = 10/0 > 1 – принимаем.

Рассмотрим альтернативы 7 и 10 (i=7,j=10):

P710 = 0 + 0 + 0 + 0 + 0 = 0;

N710 = 0 + 15 + 20 + 20 + 15 = 70;

D710 = P710/N710 = 0/70 < 1 – отбрасываем.

P107 = 0 + 15 + 20 + 20 + 15 = 70;

N107 = 0 + 0 + 0 + 0 + 0 = 0;

D107 = P107/N107 = 70/0 > 1 – принимаем.

Рассмотрим альтернативы 8 и 9 (i=8,j=9):

P89 = 0 + 0 + 0 + 20 + 10 = 30;

N89 = 0 + 10 + 0 + 0 + 0 = 10;

D89 = P89/N89 = 30/10 > 1 – принимаем.

P98 = 0 + 10 + 0 + 0 + 0 = 10;

N98 = 0 + 0 + 0 + 20 + 10 = 30;

D98 = P98/N98 = 10/30 < 1 – отбрасываем.

Рассмотрим альтернативы 8 и 10 (i=8,j=10):

P810 = 0 + 0 + 0 + 0 + 0 = 0;

N810 = 0 + 15 + 20 + 0 + 15 = 50;

D810 = P810/N810 = 0/50 < 1 – отбрасываем.

P108 = 0 + 15 + 20 + 0 + 15 = 50;

N108 = 0 + 0 + 0 + 0 + 0 = 0;

D108 = P108/N108 = 50/0 > 1 – принимаем.

Рассмотрим альтернативы 9 и 10 ($i=9,j=10$):

P910 = 0 + 0 + 0 + 0 + 0 = 0;

N910 = 0 + 15 + 20 + 20 + 15 = 70;

D910 = P910/N910 = 0/70 < 1 – отбрасываем.

P109 = 0 + 15 + 20 + 20 + 15 = 70;

N109 = 0 + 0 + 0 + 0 + 0 = 0;

D109 = P109/N109 = 70/0 > 1 – принимаем.

Составлена матрица предпочтений с внесенными и принятыми значениями D (Таблица 2.3).

Таблица 2.3 – Полная матрица предпочтений альтернатив.

	1	2	3	4	5	6	7	8	9	10
1	x				30/15	∞	∞	50/30	∞	50/35
2	40/30	x	55/25	60/30	55/15	∞	∞	∞	∞	75/15
3	45/30		x	45/30	∞	∞	∞	∞	∞	∞
4				x	50/35	50/10	∞	70/10	∞	50/15
5					x	25/15	∞	45/20	∞	
6						x	∞	45/30	∞	
7							x			
8							∞	x	30/10	
9							∞		x	
10					35/20	∞	∞	∞	∞	x

По матрице построен граф предпочтений (Рисунок 2.1).

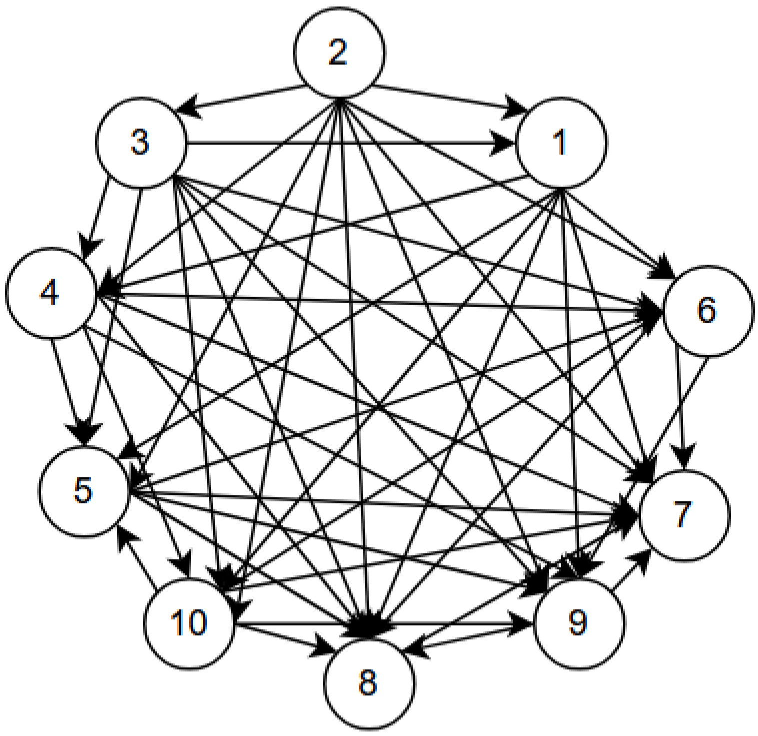

Рисунок 2.1 – Вид графа предпочтений

Назначен порог отбора предпочтений C = 1.5 (это соответствует тому, что учитываются только более сильные связи в графе).

Таким образом, матрица разрежается. В ней остаются только самые сильные связи (Таблица 2.4).

Таблица 2.4 – Матрица предпочтений проектов, при пороге С= 1.5

	1	2	3	4	5	6	7	8	9	10
1	x				30/15	∞	∞		∞	
2		x	55/25	60/30	55/15	65/10	∞	∞	∞	75/15
3	45/30		x	∞	∞	∞	∞	∞	∞	∞
4				x		50/10	∞	70/10	90/15	50/15
5					x	25/15	∞	35/20	∞	
6						x	∞	45/30	∞	
7							x			
8							∞	x		
9							∞		x	
10					35/20	35/10	∞	∞	∞	x

По этой матрице построен граф предпочтений (Рисунок 2.2).

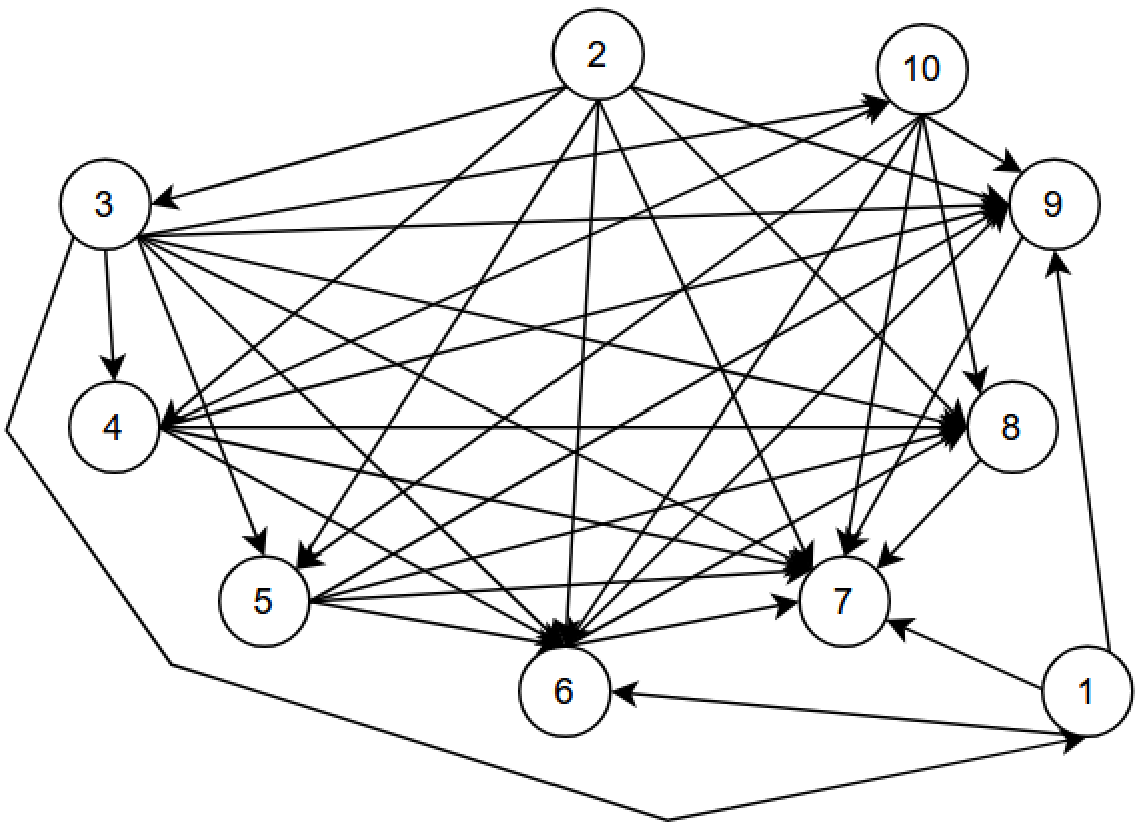

Рисунок 2.2 – Вид графа предпочтений для случая порога принятия решений С = 1.5

Петель в графе нет, при этом граф остался целостным.

Решение говорит что лучший вариант — 2. На втором месте — 3. На третьем — 10 вариант. На четвертом — 4. На пятом — 5 и 1 варианты. На шестом — 6. На седьмом — 8 и 9 варианты. На восьмом — 7 вариант.

2.3 Результат работы программы

```
Матрицы предпочтений
       0  1     2     3     4     5    6     7     8     9
0      x  -     -     - 30/15   inf  inf 50/30   inf 50/35
1  40/30  x 55/25 60/30 55/15   inf  inf   inf   inf 75/15
2  45/30  -     x 45/30   inf   inf  inf   inf   inf   inf
3      -  -     -     x 50/35 50/10  inf 70/10   inf 50/15
4      -  -     -     -     x 25/15  inf 45/20   inf     -
5      -  -     -     -     -     x  inf 45/30   inf     -
6      -  -     -     -     -     -    x     -     -     -
7      -  -     -     -     -     -  inf     x 30/10     -
8      -  -     -     -     -     -  inf     -     x     -
9      -  -     -     - 35/20 35/10  inf   inf   inf     x

Матрицы предпочтений с порогом 1.5
       0  1     2     3     4     5    6     7     8     9
0      x  -     -     - 30/15   inf  inf 50/30   inf 50/35
1  40/30  x 55/25 60/30 55/15   inf  inf   inf   inf 75/15
2  45/30  -     x 45/30   inf   inf  inf   inf   inf   inf
3      -  -     -     x 50/35 50/10  inf 70/10   inf 50/15
4      -  -     -     -     x 25/15  inf 45/20   inf     -
5      -  -     -     -     -     x  inf 45/30   inf     -
6      -  -     -     -     -     -    x     -     -     -
7      -  -     -     -     -     -  inf     x 30/10     -
8      -  -     -     -     -     -  inf     -     x     -
9      -  -     -     - 35/20 35/10  inf   inf   inf     x
```

Рисунок 2.3 – Результат работы программы. Вывод матрицы предпочтений.

2.4 Вывод по методу Электра II

В данной работе был изучен метод Электро II и использован на примере выбора самого лучшего космического корабля. Алгоритм был реализован как вручную так и программно. Данный алгоритм прост а алгоритмах, но имеет большое количество операций и сравнений. Также если выбрать слишком высокий порог, граф предпочтений превратится в лес, что увеличит количество оптимальных вариантов.

3 МЕТОД АНАЛИЗА ИЕРАРХИЙ

Метод анализа иерархий (Аналитический иерархический процесс, AHP) является одним из наиболее распространённых методов принятия решений в условиях многокритериальности. Он был разработан американским математиком Томасом Саати и основан на структурировании сложной задачи в виде иерархии целей, критериев и альтернатив. Основная идея метода заключается в попарном сравнении элементов на каждом уровне иерархии с целью определения их относительной важности. Оценки проводятся с использованием экспертных суждений, что позволяет учитывать как количественные, так и качественные характеристики. Результатом метода является взвешенная оценка альтернатив и выбор наилучшего варианта. Благодаря своей гибкости и наглядности, метод широко применяется в экономике, управлении, проектировании и других сферах.

3.1 Постановка задачи

Задача практической работы: выбрать лучший космический корабль.

3.2 Представление проблемы в виде иерархии

Первый этап – представление проблемы в виде иерархии или сети. В простейшем случае, иерархия строится, начиная с цели, которая помещается в вершину иерархии. Через промежуточные уровни, на которых располагаются критерии и от которых зависят последующие уровни, к самому низкому уровню, который содержит перечень альтернатив.

Иерархия считается полной, если каждый элемент заданного уровня является критерием для всех элементов нижнего уровня

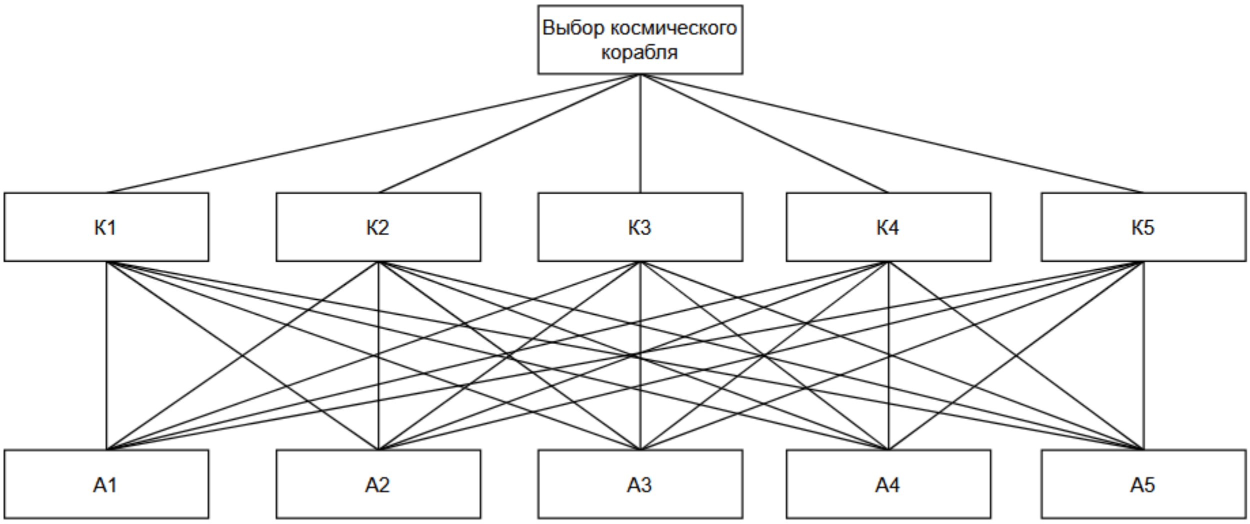

Рисунок 3.1 – Полная доминантная иерархия.

Критерии:

К 1 – Цена;

К 2 – Скорость;

К 3 – Время входа в гиперпространство;

К 4 – Кол-во орудий;

К 5 – Мощность щитов.

Альтернативы:

А 1 – TIE Fighter;

А 2 – S-100;

А 3 – CR90;

А 4 – FT-6;

А 5 – S-13.

3.3 Установка приоритетов критериев

Таблица данных:

Таблица.3.1.1 – Шкала относительной важности.

	Космический корабль	Цена (кредиты) (-)	Скорость (км/ч) (+)	Время входа в гиперпространство (сек) (-)	Количество орудий (шт) (+)	Мощность щитов (Вт) (+)
1	TIE Fighter	20000	5000	3.0	2	100
2	S-100	21000	4800	3.1	3	150
3	CR90	25000	4600	3.5	2	130
4	FT-6	35000	4400	4.5	2	100
5	S-13	33000	4600	4.1	2	105

После иерархического представления задачи установлены приоритеты критериев и оценена каждая из альтернатив по критериям, определена наиболее важная их них. В методе анализа иерархий элементы сравниваются попарно по отношению к их влиянию на общую для них характеристику. Парные сравнения приводят к записи характеристик сравнений в виде квадратной таблицы чисел, которая называется матрицей. Для облегчения работы введена шкала относительной важности (Таблица 3.1.2).

Таблица.3,1.2 – Шкала относительной важности.

Интенсивность относительной важности	Определение	Объяснение
1	Равная важность	Равный вклад двух критериев в цель.
3	Слабое превосходство	Дают легкое превосходство одной альтернативы над другой
5	Умеренное превосходство	Опыт и суждения дают умеренное превосходство
7	Сильное превосходство	Одному из критериев дается настолько сильное предпочтение.
9	Абсолютное превосходство	Очевидность превосходства одного критерия над другим
2,4,6,8	Промежуточные решения между двумя соседними суждениями	Применяется в компромиссных случаях

Шкала содержит соответствующие обратные значения.

3.4 Синтез приоритетов

После построения иерархии и определения величин парных субъективных суждений следует этап, на котором иерархическая декомпозиция и относительные суждения объединяются для получения осмысленного решения многокритериальной задачи принятия решений. Из групп парных сравнений формируется набор локальных критериев, которые выражают относительное влияние элементов на элемент, расположенный на уровне выше. Составлена обратно симметричная матрица для парного сравнения критериев (Таблица 2).

Таблица 3.2 – Матрица парного сравнения критериев.

Цель	К 1	К 2	К 3	К 4	К 5	Vi	W_{2i}
К 1	1	1/7	3	2	2	1,113	0,170
К 2	7	1	3	5	5	3,500	0,534
К 3	1/3	1/3	1	1	1	0,644	0,098
К 4	1/2	1/5	1	1	3	0,786	0,120
К 5	1/2	1/5	1	1/3	1	0,506	0,077
∑Vi						6,549	

Для определения относительной ценности каждого элемента необходимо найти геометрическое среднее и с этой целью перемножить n элементов каждой строки и из полученного результата извлечь корни n-й степени (размерность матрицы n=5).

Строка № 1

$V_1=(1 \times 1/7 \times 3 \times 2 \times 2)^{1/5} = 1{,}113$;

Строка № 2

$V_2=(7 \times 1 \times 3 \times 5 \times 5\ )^{1/5} = 3{,}500$;

Строка № 3

$V_3=(\ 1/3 \times 1/3 \times 1 \times 1 \times 1)^{1/5} = 0{,}644$;

Строка № 4

$V_4=(\ 1/2 \times 1/5 \times 1 \times 1 \times 3)^{1/5} = 0{,}786$;

Строка № 5

$V_5=(\ 1/2 \times 1/5 \times 1 \times 1/3 \times 1)^{1/5} = 0{,}506$.

Проведена нормализация полученных чисел. Для этого определен нормирующий коэффициент ∑Vi.

$\sum Vi = V_1 + V_2 + V_3 + V_4 + V_5 = 1{,}113 + 3{,}500 + 0{,}644 + 0{,}786 + 0{,}506 = 6{,}549$.

Найдена важность приоритетов W_{2i}, для этого каждое из чисел Vi разделено на ∑Vi.

Строка № 1

W_{21}= 1,113 / $\sum Vi$ =Y_{21} = 0,170;

Строка № 2

W_{22}= 3,500 / $\sum Vi$ = Y_{22} = 0,534;

Строка № 3

W_{23}= 0,644 / $\sum Vi$ = Y_{23} = 0,098;

Строка № 4

W_{24}= 0,786 / $\sum Vi$ = Y_{24} = 0,120;

Строка № 5

W_{25}= 0,506 / $\sum Vi$ = Y_{25} = 0,077.

В результате получен вектор приоритетов:

W_{2i} = (0,170 = Y_{21}; 0,534 = Y_{22}; 0,098 = Y_{23}; 0,120 = Y_{24}; 0,077 = Y_{25}), где индекс 2 означает, что вектор приоритетов относится ко второму уровню иерархии.

К 1 – цена(Таблица 3);

Таблица 3.3 – Матрица сравнения по критерию 1.

К1	А1	А2	А3	А4	А5	V_{K1Y}	W_{3K1Y}
А1	1	3	3	5	5	2,954	0,436
А2	1/3	1	3	5	5	1,904	0,281
А3	1/3	1/3	1	5	3	1,107	0,164
А4	1/5	1/5	1/5	1	1	0,380	0,056
А5	1/5	1/5	1/3	1	1	0,422	0,062
$\sum V_{K1Y}$						6,767	

Определена относительная ценность каждого элемента.

Строка № 1

V_{K11}=(1 x 3 x 3 x 5 x 5)$^{1/5}$= 2,954;

Строка № 2

V_{K12}=(1/3 x 1 x 3 x 5 x 5)$^{1/5}$= 1,904;

Строка № 3

V_{K13}=(1/3 x 1/3 x 1 x 5 x 3)$^{1/5}$= 1,107;

Строка № 4

V_{K14}=(1/5 x 1/5 x 1/5 x 1 x 1)$^{1/5}$= 0,380;

Строка № 5

V_{K15}=(1/5 x 1/5 x 1/3 x 1 x 1)$^{1/5}$= 0,422.

Проведена нормализация полученных чисел. Для этого определен нормирующий коэффициент $\sum V_{K1Y}$.

$\sum V_{K1Y}=V_{K11}+V_{K12}+V_{K13}+V_{K14}+V_{K15}$=2,954+1,904+1,107+0,380+0,422 = 6,767.

Найдена важность приоритетов W_{3K1Y}, для этого каждое из чисел V_{K1Y} разделено на $\sum V_{K1Y}$.

Строка № 1

W_{3K11}= 2,954 / $\sum$Vi = Y_{311} = 0,436

Строка № 2

W_{3K12}= 1,904 / $\sum$Vi = Y_{312} = 0,281

Строка № 3

W_{3K13}= 1,107 / $\sum$Vi = Y_{313} = 0,164

Строка № 4

W_{3K14}= 0,380 / $\sum$Vi = Y_{314} = 0,056

Строка № 5

W_{3K15}= 0,422 / $\sum$Vi = Y_{315} = 0,062

В результате получаем вектор приоритетов:

W_{3K1Y} = (Y_{311} = 0,436; Y_{312} = 0,281; Y_{313} = 0,164; Y_{314} = 0,056; Y_{315} = 0,062),

где индекс 3 означает, что вектор приоритетов относится к третьему уровню иерархии критерия К1.

К 2 – скорость(Таблица 4);

Таблица 3.4. – Матрица сравнения по критерию 2.

К2	А1	А2	А3	А4	А5	V_{K2Y}	W_{3K2Y}
А1	1	1/3	1/3	1/5	1/3	0,375	0,064
А2	3	1	1/3	1/3	1/3	0,644	0,110
А3	3	3	1	1	1	1,552	0,266
А4	5	3	1	1	1	1,719	0,294
А5	3	3	1	1	1	1,552	0,266
$\sum V_{K2Y}$						5,842	

Определена относительная ценность каждого элемента.

Строка № 1

V_{K21}=(1 x 1/3 x 1/3 x 1/5 x 1/3)$^{1/5}$= 0,375;

Строка № 2

V_{K22}=(3 x 1 x 1/3 x 1/3 x 1/3)$^{1/5}$= 0,644;

Строка № 3

V_{K23}=(3 x 3 x 1 x 1 x 1)$^{1/5}$= 1,552;

Строка № 4

V_{K24}=(5 x 3 x 1 x 1 x 1)$^{1/5}$= 1,719;

Строка № 5

V_{K25}=(3 x 3 x 1 x 1 x 1)$^{1/5}$= 1,552.

Проведена нормализация полученных чисел. Для этого определен нормирующий коэффициент $\sum V_{K2Y}$.

$\sum V_{K2Y} = V_{K21} + V_{K22} + V_{K23} + V_{K24} + V_{K25}$ = 0,375+0,644+1,552+1,719+1,552 = 5,842.

Найдена важность приоритетов W_{3K2Y}, для этого каждое из чисел V_{K2Y} разделено на $\sum V_{K2Y}$.

Строка № 1

W_{3K21}= 0,375 / $\sum$Vi = 0,064;

Строка № 2

W_{3K22}= 0,644 / $\sum$Vi = 0,110;

Строка № 3

W_{3K23}= 1,552 / $\sum$Vi = 0,266;

Строка № 4

W_{3K24}= 1,719 / $\sum$Vi = 0,294;

Строка № 5

W_{3K25}= 1,552 / $\sum$Vi = 0,266.

В результате получаем вектор приоритетов:

W_{3K2Y} = (Y_{321} = 0,064; Y_{322} =0,110; Y_{323} =0,266; Y_{324} =0,294; Y_{325} =0,266),

где индекс 3 означает, что вектор приоритетов относится к третьему уровню иерархии критерия К2.

К 3 – время входа в гиперпространство (Таблица 5);

Таблица 3.5 – Матрица сравнения по критерию 5.

К3	А1	А2	А3	А4	А5	V_{K3Y}	W_{3K3Y}
А1	1	1	3	5	5	2,371	0,372
А2	1	1	1	5	5	1,904	0,299
А3	1/3	1	1	3	3	1,246	0,195
А4	1/5	1/5	1/3	1	1	0,422	0,066
А5	1/5	1/5	1/3	1	1	0,422	0,066
V_{K35}						6,365	

Определена относительная ценность каждого элемента.

Строка № 1

V_{K31}=(1 x 1 x 3 x 5 x 5)$^{1/5}$= 2,371;

Строка № 2

V_{K32}=(1 x 1 x 1 x 5 x 5)$^{1/5}$= 1,904;

Строка № 3

V_{K33}=(1/3 x 1 x 1 x 3 x 3)$^{1/5}$= 1,246;

Строка № 4

V_{K34}=(1/5 x 1/5 x 1/3 x 1 x 1)$^{1/5}$= 0,422;

Строка № 5

V_{K35}=(1/5 x 1/5 x 1/3 x 1 x 1)$^{1/5}$= 0,422.

Проведена нормализация полученных чисел. Для этого определен нормирующий коэффициент $\sum V_{K3Y}$.

$\sum V_{K3Y} = V_{K31} + V_{K32} + V_{K33} + V_{K34} + V_{K35}$ = 2,371+1,904+1,246+0,422+0,422 = 6,365.

Найдена важность приоритетов W_{3K2Y}, для этого каждое из чисел V_{K2Y} разделено на $\sum V_{K2Y}$.

Строка № 1

W_{3K31}= 2,371 / $\sum$Vi = 0,372;

Строка № 2

W_{3K32}= 1,904 / $\sum$Vi = 0,299;

Строка № 3

W_{3K33}= 1,246 / $\sum$Vi = 0,195;

Строка № 4

W_{3K34}= 0,422 / $\sum$Vi = 0,066;

Строка № 5

W_{3K35}= 0,422 / ∑Vi = 0,066.

В результате получаем вектор приоритетов:

W_{3K3Y} = (Y_{331}=0,372; Y_{332}=0,299; Y_{333}=0,195; Y_{334}=0,066; Y_{335}=0,066),

где индекс 3 означает, что вектор приоритетов относится к третьему уровню иерархии критерия К3.

К 4 – кол-во орудий (Таблица 6);

Таблица 3.6 – Матрица сравнения по критерию 4.

К4	А1	А2	А3	А4	А5	V_{K4Y}	W_{3K4Y}
А1	1	3	1	1	1	1,246	0,312
А2	1/3	1	1/3	1/3	1/3	0,333	0,083
А3	1	1/3	1	1	1	0,803	0,201
А4	1	1/3	1	1	1	0,803	0,201
А5	1	1/3	1	1	1	0,803	0,201
$\sum V_{K4Y}$						3,988	

Определена относительная ценность каждого элемента.

Строка № 1

V_{K41}=(1 x 3 x 1 x 1 x 1)$^{1/5}$ = 1,246;

Строка № 2

V_{K42}=(1/3 x 1/3 x 1/3 x 1/3 x 1/3)$^{1/5}$ =0,333 ;

Строка № 3

V_{K43}=(1 x 1/3 x 1 x 1 x 1)$^{1/5}$ = 0,803;

Строка № 4

V_{K44}=(1 x 1/3 x 1 x 1 x 1)$^{1/5}$ = 0,803;

Строка № 5

V_{K45}=(1 x 1/3 x 1 x 1 x 1)$^{1/5}$ = 0,803.

Проведена нормализация полученных чисел. Для этого определен нормирующий коэффициент $\sum V_{K4Y}$.

$\sum V_{K4Y} = V_{K41} + V_{K42} + V_{K43} + V_{K44} + V_{K45}$ =1,246+0,333+0,803+0,803+0,803 = 3,988 .

Найдена важность приоритетов W_{3K4Y}, для этого каждое из чисел V_{K4Y} разделено на $\sum V_{K4Y}$.

Строка № 1

W_{3K41}= 1,246 / $\sum Vi$ =0,312 ;

Строка № 2

W_{3K42}= 0,333 / $\sum Vi$ = 0,083;

Строка № 3

W_{3K43}= 0,803 / $\sum Vi$ = 0,201;

Строка № 4

W_{3K44}= 0,803 / $\sum Vi$ = 0,201;

Строка № 5

W_{3K45}= 0,803 / $\sum Vi$ = 0,201.

В результате получаем вектор приоритетов:

W_{3K4Y} = (Y_{341}=0,312; Y_{342}=0,083; Y_{343}=0,201; Y_{344}=0,201; Y_{345}=0,201),

где индекс 3 означает, что вектор приоритетов относится к третьему уровню иерархии критерия К4.

К 5 – мощность шитов (Таблица 7).

Таблица 3.7 – Матрица сравнения по критерию 5.

К5	А1	А2	А3	А4	А5	V_{K5Y}	W_{3K5Y}
А1	1	5	5	3	3	2,954	0,454
А2	1/5	1	3	5	5	1,719	0,264
А3	1/5	1/3	1	3	3	0,902	0,138
А4	1/3	1/5	1/3	1	1	0,467	0,072
А5	1/3	1/5	1/3	1	1	0,467	0,072
$\sum V_{K5Y}$						6,509	

Определена относительная ценность каждого элемента.

Строка № 1

V_{K51}=(1 x 5 x 5 x 3 x 3)$^{1/5}$ = 2,954;

Строка № 2

V_{K52}=(1/5 x 1 x 3 x 5 x 5)$^{1/5}$ = 1,719;

Строка № 3

V_{K53}=(1/5 x 1/3 x 1 x 3 x 3)$^{1/5}$ = 0,902;

Строка № 4

V_{K54}=(1/3 x 1/5 x 1/3 x 1 x 1)$^{1/5}$ = 0,467;

Строка № 5

V_{K55}=(1/3 x 1/5 x 1/3 x 1 x 1)$^{1/5}$ = 0,467.

Проведена нормализация полученных чисел. Для этого определен нормирующий коэффициент $\sum V_{K5Y}$.

$\sum V_{K5Y} = V_{K51} + V_{K52} + V_{K53} + V_{K54} + V_{K55} =$ 2,954+1,719+0,902+0,467+0,467=6,509.

Найдена важность приоритетов W_{3K5Y}, для этого каждое из чисел V_{K5Y} разделено на $\sum V_{K5Y}$.

Строка № 1

W_{3K51}= 2,954 / $\sum$Vi = 0,454;

Строка № 2

W_{3K52}= 1,719 / $\sum$Vi = 0,264;

Строка № 3

W_{3K53}= 0,902 / $\sum$Vi = 0,138;

Строка № 4

W_{3K54}= 0,467 / $\sum$Vi = 0,072;

Строка № 5

W_{3K55}= 0,467 / ∑Vi = 0,072.

В результате получаем вектор приоритетов:

W_{3K5Y} = (Y_{351}=0,454; Y_{352}=0,264; Y_{353}=0,138; Y_{354}=0,072; Y_{355}=0,072),

где индекс 3 означает, что вектор приоритетов относится к третьему уровню иерархии критерия К5.

3.5 Согласованность локальных приоритетов

Любая матрица суждений в общем случае не согласована, так как суждения отражают субъективные мнения ЛПР, а сравнение элементов, которые имеют количественные эквиваленты, может быть несогласованным из-за присутствия погрешности при проведении измерений. Совершенной согласованности парных сравнений даже в идеальном случае на практике достичь трудно. Нужен способ оценки степени согласованности при решении конкретной задачи.

Метод анализа иерархий дает возможность провести такую оценку.

Вместе с матрицей парных сравнений есть мера оценки степени отклонения от согласованности. Когда такие отклонения превышают установленные пределы тем, кто проводит решение задачи, необходимо их пересмотреть.

В таблице приведены средние значения индекса случайной согласованности (СИ) для случайных матриц суждений разного порядка.

В нашей задаче размерность матрицы n=5, тогда среднее значение индекса случайной согласованности СИ = 1,12.

Определены индекс согласованности и отношение согласованности для матрицы «Выбор космического коробля» (Таблица 8).

Таблица 3.8 – Матрица «Выбор лучшего отеля».

Цель	К 1	К 2	К 3	К 4	К 5	W_{2i}
К 1	1	1/7	3	2	2	0,170
К 2	7	1	3	5	5	0,534
К 3	1/3	1/3	1	1	1	0,098
К 4	1/2	1/5	1	1	3	0,120
К 5	1/2	1/5	1	1/3	1	0,077

Определена сумма каждого столбца матрицы суждений.

$S_1 = 1 + 7 + 1/3 + 1/2 + 1/2 = 9{,}3$;

$S_2 = 1/7 + 1 + 1/3 + 1/5 + 1/5 = 1{,}876$;

$S_3 = 3 + 3 + 1 + 1 + 1 = 9$;

$S_4 = 2 + 5 + 1 + 1 + 1/3 = 9{,}333$;

$S_5 = 2 + 5 + 1 + 3 + 1 = 12$.

Полученный результат умножен на компоненту нормализованного вектора приоритетов, т.е. сумму суждений первого столбца на первую компоненту, сумму суждений второго столбца - на вторую и т.д.

$P_1 = S_1 \times W_{21} = 9{,}3 \times 0{,}170 = 1{,}581$;

$P_2 = S_2 \times W_{22} = 1{,}876 \times 0{,}534 = 1{,}001$;

$P_3 = S_3 \times W_{23} = 9 \times 0{,}098 = 0{,}882$;

$P_4 = S_4 \times W_{24} = 9{,}333 \times 0{,}120 = 1{,}120$;

$P_5 = S_5 \times W_{25} = 12 \times 0{,}077 = 0{,}824$.

Сумма чисел Pj отражает пропорциональность предпочтений, чем ближе эта величина к n (числу объектов и видов действия в матрице парных сравнений), тем более согласованны суждения.

$$\lambda_{max} = P_1 + P_2 + P_3 + P_4 + P_5 = 5{,}405.$$

Отклонение от согласованности выражается индексом согласованности.

$$ИС = (\lambda_{max} - n)/(n - 1) = 0{,}405/4 = 0{,}101.$$

Отношение индекса согласованности ИС к среднему значению случайного индекса согласованности СИ называется отношением согласованности ОС.

$$ОС = ИС/СИ = 0{,}101/1{,}12 = 0{,}090.$$

Значение ОС меньше или равное 0.10 считается приемлемым, значит матрица «Выбор лучшего спортсмена» согласована.

Определнены индекс согласованности и отношение согласованности для матрицы К 1 – цена(Таблица 3.9).

Таблица 3.9 – Матрица сравнения по критерию 1.

К1	А1	А2	А3	А4	А5	W_{3K1Y}
А1	1	3	3	5	5	0,436
А2	1/3	1	3	5	5	0,281
А3	1/3	1/3	1	5	3	0,164
А4	1/5	1/5	1/5	1	1	0,056
А5	1/5	1/5	1/3	1	1	0,062

Определяется сумма каждого столбца матрицы суждений.

$S_{1К1} = 2.666$;

$S_{2\,К1} = 5.333$;

$S_{3\,К1} = 7.833$;

$S_{4\,К1} = 17$;

$S_{5\,К1} = 15$.

Затем полученный результат умножен на компоненту нормализованного вектора приоритетов.

$P_{1\,К1} = S_1 \text{ x } W_{3К11} = 1.133$;

$P_{2\,К1} = S_2 \text{ x } W_{3К12} = 1.489$;

$P_{3\,К1} = S_3 \text{ x } W_{3К13} = 1.284$;

$P_{4\,К1} = S_1 \text{ x } W_{3К14} = 0.652$;

$P_{5\,К1} = S_1 \text{ x } W_{3К15} = 0.93$.

Найдена пропорциональность предпочтений.

$$\lambda_{max\,К1} = P_{1К1} + P_{2К1} + P_{3К1} + P_{4К1} + P_{5К1} = 5.448.$$

Отклонение от согласованности выражается индексом согласованности.

$$ИС_{К1} = (\lambda_{max\,К1} - n)/(n - 1) = 0.112.$$

Найдено отношение согласованности ОС.

$$ОС_{К1} = ИС/СИ = 0.1.$$

Значение ОС меньше или равное 0.10 считается приемлемым, значит матрица К 1 (цена) согласована.

Определены индекс согласованности и отношение согласованности для матрицы К 2 – скорость (Таблица 3.10).

Таблица 3.10 – Матрица сравнения по критерию 2.

К2	А1	А2	А3	А4	А5	W_{3K2Y}
А1	1	1/3	1/3	1/5	1/3	0,064
А2	3	1	1/3	1/3	1/3	0,110
А3	3	3	1	1	1	0,266
А4	5	3	1	1	1	0,294
А5	3	3	1	1	1	0,266

Определена сумма каждого столбца матрицы суждений.

$S_{1К2}$ =17;

$S_{2\,К2}$ =10.3;

$S_{3\,К2}$ =3.6;

$S_{4\,К2}$ =3.8;

$S_{5\,К2}$ =3.6.

Затем полученный результат умножен на компоненту нормализованного вектора приоритетов.

$P_{1\,К2} = S_1 \times W_{3\,К21}$ =1.088;

$P_{2\,К2} = S_2 \times W_{3\,К22}$ =1.133;

$P_{3\,К2} = S_3 \times W_{3\,К23}$ =0.957;

$P_{4\,К2} = S_4 \times W_{3\,К24}$ =1.117;

$P_{5\,К2} = S_5 \times W_{3\,К25}$ =0.957.

Найдена пропорциональность предпочтений.

$$\lambda_{max\,К2} = P_{1К2} + P_{2К2} + P_{3К2} + P_{4К2} + P_{5К2} = 5.252.$$

Отклонение от согласованности выражается индексом согласованности.

$$ИС_{\,К2} = (\lambda_{max\,К2} - n)/(n - 1) = 0.063.$$

Найдено отношение согласованности ОС.

$$ОС_{\,К2} = ИС/СИ = 0.056.$$

Значение ОС меньше или равное 0.10 считается приемлемым, значит матрица К 2 (скорость) согласована.

Определены индекс согласованности и отношение согласованности для матрицы К 3 – время входа в гиперпространство (Таблица 3.11).

Таблица 3.11 – Матрица сравнения по критерию 3.

К3	А1	А2	А3	А4	А5	$W_{3К3Y}$
А1	1	1	3	5	5	0,372
А2	1	1	1	5	5	0,299
А3	1/3	1	1	3	3	0,195
А4	1/5	1/5	1/3	1	1	0,066
А5	1/5	1/5	1/3	1	1	0,066

Определена сумма каждого столбца матрицы суждений.

$S_{1К3} = 3.3$;

$S_{2\,К3} = 4$;

$S_{3\,К3} = 5.6$;

$S_{4\,К3} = 15$;

$S_{5\,К3} = 15$.

Затем полученный результат умножен на компоненту нормализованного вектора приоритетов.

$P_{1\,К3} = S_1 \text{ x } W_{3\,К31} = 1.227$;

$P_{2\,К3} = S_2 \text{ x } W_{3\,К32} = 1.196$;

$P_{3\,К3} = S_3 \text{ x } W_{3\,К33} = 1.092$;

$P_{4\,К3} = S_4 \text{ x } W_{3\,К34} = 0.99$;

$P_{5\,К3} = S_5 \text{ x } W_{3\,К35} = 0.99$.

Найдем пропорциональность предпочтений.

$$\lambda_{max\ К3} = P_{1К3} + P_{2К3} + P_{3К3} + P_{4К3} + P_{5К3} = 5.495.$$

Отклонение от согласованности выражается индексом согласованности.

$$ИС_{К3} = (\lambda_{max\ К3} - n)/(n - 1) = 0.123.$$

Найдено отношение согласованности ОС.

$$ОС_{К3} = ИС/СИ = 0.10.$$

Значение ОС меньше или равное 0.10 считается приемлемым, значит матрица К 3 (время входа в гиперпространство) согласована.

Определены индекс согласованности и отношение согласованности для матрицы К 4 – кол-во орудий (Таблица 3.12).

Таблица 3.12 – Матрица сравнения по критерию 4.

К4	А1	А2	А3	А4	А5	W_{3K4Y}
А1	1	3	1	1	1	0,312
А2	1/3	1	1/3	1/3	1/3	0,083
А3	1	1/3	1	1	1	0,201
А4	1	1/3	1	1	1	0,201
А5	1	1/3	1	1	1	0,201

Определена сумма каждого столбца матрицы суждений.

S_{1K4} =4.3;

S_{2K4} =5;

S_{3K4} =4.3;

S_{4K4} =4.3;

S_{5K4} =4.3.

Затем полученный результат умножен на компоненту нормализованного вектора приоритетов.

$P_{1K4} = S_1 \text{ x } W_{3\ K41}$ =1.341;

$P_{2K4} = S_2 \text{ x } W_{3\ K42}$ =0.415;

$P_{3K4} = S_3 \text{ x } W_{3\ K43}$ =0.864;

$P_{4K4} = S_4 \text{ x } W_{3\ K44}$ =0.864;

$P_{5K4} = S_5 \text{ x } W_{3\ K45}$ =0.864.

Найдена пропорциональность предпочтений.

$$\lambda_{max\ K4} = P_{1K4} + P_{2K4} + P_{3K4} + P_{4K4} + P_{5K4} = 4.348.$$

Отклонение от согласованности выражается индексом согласованности.

$$ИС_{\ K4} = (\lambda_{max\ K4} - n)/(n - 1) = 0.087.$$

Найдено отношение согласованности ОС.

$$ОС_{\ K4} = ИС/СИ = 0.077.$$

Значение ОС меньше или равное 0,10 считается приемлемым, значит матрица К 4 (кол-во орудий) согласована.

Определены индекс согласованности и отношение согласованности для матрицы К 5 – мощность щитов (Таблица 3.13).

Таблица 3.13 – Матрица сравнения по критерию 5.

К5	А1	А2	А3	А4	А5	W_{3K5Y}
А1	1	5	5	3	3	0,454
А2	1/5	1	3	5	5	0,264
А3	1/5	1/3	1	3	3	0,138
А4	1/3	1/5	1/3	1	1	0,072
А5	1/3	1/5	1/3	1	1	0,072

Определена сумма каждого столбца матрицы суждений.

$S_{1К5}$ =2.6;

$S_{2К5}$ =7.3;

$S_{3К5}$ =9.6;

$S_{4К5}$ =13;

$S_{5К5}$ =13.

Затем полученный результат умножен на компоненту нормализованного вектора приоритетов.

$P_{1К5} = S_1$ x $W_{3\ К41}$ =0.180;

$P_{2К5} = S_2$ x $W_{3\ К42}$ =1.927;

$P_{3К5} = S_3$ x $W_{3\ К43}$ =1.296;

$P_{4К5} = S_1$ x $W_{3\ К44}$ =0.936;

$P_{5К5} = S_1$ x $W_{3\ К45}$ =0.936.

Найдена пропорциональность предпочтений.

$$\lambda_{max\ К5} = P_{1К5} + P_{2К5} + P_{3К5} + P_{4К5} + P_{5К5} = 5.275.$$

Отклонение от согласованности выражается индексом согласованности.

$$ИС_{\ К5} = (\lambda_{max\ К5} - n)/(n - 1) = 0.068.$$

Найдено отношение согласованности ОС.

$$ОС_{\ К5} = ИС/СИ = 0.061.$$

Значение ОС меньше или равное 0,10 считается приемлемым, значит матрица К 5 (мощность щитов) согласована.

3.6 Синтез альтернатив

Векторы приоритетов и отношения согласованности определяются для всех матриц суждений, начиная со второго уровня.

Для определения приоритетов альтернатив локальные приоритеты умножены на приоритет соответствующего критерия на высшем уровне и найдены суммы по каждому элементу в соответствии с критериями, на которые воздействует этот элемент.

$W_{2i} = (0{,}170 = Y_{21}; 0{,}534 = Y_{22}; 0{,}098 = Y_{23}; 0{,}120 = Y_{24}; 0{,}077 = Y_{25})$;

$W_{3K1Y} = (Y_{311} = 0{,}436; Y_{312} = 0{,}281; Y_{313} = 0{,}164; Y_{314} = 0{,}056; Y_{315} = 0{,}062)$;

$W_{3K2Y} = (Y_{321} = 0{,}064; Y_{322} = 0{,}110; Y_{323} = 0{,}266; Y_{324} = 0{,}294; Y_{325} = 0{,}266)$;

$W_{3K3Y} = (Y_{331}=0{,}372; Y_{332}=0{,}299; Y_{333}=0{,}195; Y_{334}=0{,}066; Y_{335}=0{,}066)$;

$W_{3K4Y} = (Y_{341}=0{,}312; Y_{342}=0{,}083; Y_{343}=0{,}201; Y_{344}=0{,}201; Y_{345}=0{,}201)$;

$W_{3K5Y} = (Y_{351}=0{,}454; Y_{352}=0{,}264; Y_{353}=0{,}138; Y_{354}=0{,}072; Y_{355}=0{,}072)$.

Приоритеты альтернатив получены следующим образом:

$W_1 = W_{21} \text{ x } W_{3K11} + W_{22} \text{ x } W_{3K21} + W_{23} \text{ x } W_{3K31} + W_{24} \text{ x } W_{3K41} + W_{25} \text{ x } W_{3K51} =0{,}216$.

$W_2 = W_{21} \text{ x } W_{3K12} + W_{22} \text{ x } W_{3K22} + W_{23} \text{ x } W_{3K32} + W_{24} \text{ x } W_{3K42} + W_{25} \text{ x } W_{3K52} =0{,}162$.

$W_3 = W_{21} \text{ x } W_{3K13} + W_{22} \text{ x } W_{3K23} + W_{23} \text{ x } W_{3K33} + W_{24} \text{ x } W_{3K43} + W_{25} \text{ x } W_{3K53} =0{,}221$

$W_4 = W_{21} \text{ x } W_{3K14} + W_{22} \text{ x } W_{3K24} + W_{23} \text{ x } W_{3K34} + W_{24} \text{ x } W_{3K44} + W_{25} \text{ x } W_{3K54} =0{,}119$.

$W_5 = W_{21} \text{ x } W_{3K15} + W_{22} \text{ x } W_{3K25} + W_{23} \text{ x } W_{3K35} + W_{24} \text{ x } W_{3K45} + W_{25} \text{ x } W_{3K55} =0{,}201$.

Таким образом, приоритеты альтернатив равны:

альтернатива А1 (название) - W_1 приоритет равен =0,216;

альтернатива А2 (название)- W_2 приоритет равен =0,162;

альтернатива А3 (название) - W_3 приоритет равен =0,221;

альтернатива А4 (название) – W_4 приоритет равен =0,119;

альтернатива А5 (название) - W_5 приоритет равен =0,201.

Самая оптимальная альтернатива имеет имеет наибольший приоритет, это альтернатива №3.

3.7 Результаты работы программы

```
Вектор приоритетов [0.17, 0.534, 0.098, 0.12, 0.077]
Вектор приоритетов [0.436, 0.281, 0.164, 0.056, 0.062]
Вектор приоритетов [0.063, 0.098, 0.261, 0.289, 0.289]
Вектор приоритетов [0.373, 0.299, 0.196, 0.066, 0.066]
Вектор приоритетов [0.306, 0.102, 0.197, 0.197, 0.197]
Вектор приоритетов [0.454, 0.264, 0.139, 0.072, 0.072]

Приоритет альтернативы №1 : 0.216
Приоритет альтернативы №2 : 0.162
Приоритет альтернативы №3 : 0.221
Приоритет альтернативы №4 : 0.199
Приоритет альтернативы №5 : 0.201

Лучшая альтернатива : №3
```

Рисунок 3.2 – Вывод программы

3.8 Вывод по методу МАИ

В ходе данной работы изучен метод анализа иерархий, проведён его ручной расчёт для пяти критериев и пяти альтернатив. Преимуществом метода является гарантированное получение единственного оптимального решения, а недостатком является требование соблюдать согласованность матриц приоритетов, из-за чего необходимо проводить повторные расчёты в случае, если матрица не согласована.

4 ГРАФИЧЕСКИЙ МЕТОД

Линейное программирование является важным разделом математического программирования, позволяющим находить оптимальные решения задач, связанных с распределением ограниченных ресурсов. Одним из наиболее наглядных и интуитивно понятных методов решения задач линейного программирования является графический метод.

Графический метод применяется в случаях, когда число переменных в задаче не превышает двух, поскольку его суть заключается в геометрической интерпретации системы ограничений и нахождении оптимального значения целевой функции на построенной области допустимых решений. Метод позволяет не только получить решение, но и визуально проанализировать, какие ограничения оказывают влияние на результат, а также оценить чувствительность оптимального решения к изменению параметров задачи.

Область применения графического метода охватывает широкий спектр задач, связанных с управлением ресурсами, планированием производства, оптимизацией затрат и максимизацией прибыли. В частности, он используется для решения задач оптимального использования сырья в производстве, распределения финансовых средств, планирования логистических маршрутов и управления запасами.

4.1 Постановка задачи

Решить задачу линейного программирования с двумя переменными графическим методом.

4.2 Данные индивидуального варианта

$$f(x) = -2x_1 + x_2 \to min/max$$

$$\begin{cases} 4x_1 - x_2 \geq -4 \\ 2x_1 + 3x_2 \leq 12 \\ 5x_1 - 3x_2 \leq 15 \\ x_1, x_2 \geq 0 \end{cases}$$

4.3 Подготовка данных

В среде Microsoft Excel добавим 5 столбцов:

1. x_1 – значения от 0 до 10 с шагом 0,5;
2. $x_2 = (\dots)$ – значения ограничения $(4x1 - x2 \geq -4)$;
3. $x_2 = (\dots)$ – значения ограничения $(2x1 + 3x2 \leq 12)$;
4. $x_2 = (\dots)$ – значения ограничения $(5x1 - 3x2 \leq 15)$;
5. $x_2 = (\dots)$ – значения целевой функции при условии $f(x)$ = 0.

Таблица 4,1.1 – Данные для графика

x1	x2= 4 + 4x1	x2 = (12-2x1)/3	x2=(5x1-15)/3	x2=2x1
0	4	4	-5	0
0,5	6	3,67	-4,17	1
1	8	3,33	-3,33	2
1,5	10	3	-2,5	3
2	12	2,67	-1,67	4
2,5	14	2,33	-0,83	5
3	16	2	0	6
3,5	18	1,67	0,83	7
4	20	1,33	1,67	8
4,5	22	1	2,5	9
5	24	0,67	3,33	10
5,5	26	0,33	4,17	11
6	28	0	5	12
6,5	30	-0,33	5,83	13
7	32	-0,67	6,67	14
7,5	34	-1	7,5	15
8	36	-1,33	8,33	16
8,5	38	-1,67	9,17	17
9	40	-2	10	18
9,5	42	-2,33	10,83	19
10	44	-2,67	11,67	20

4.4 Построение графика

Выделим таблицу подготовленных данных и построим гладкий график. Произведем настройку шага координатной оси x_1 и получим следующий график (Рисунок 4.1.1)

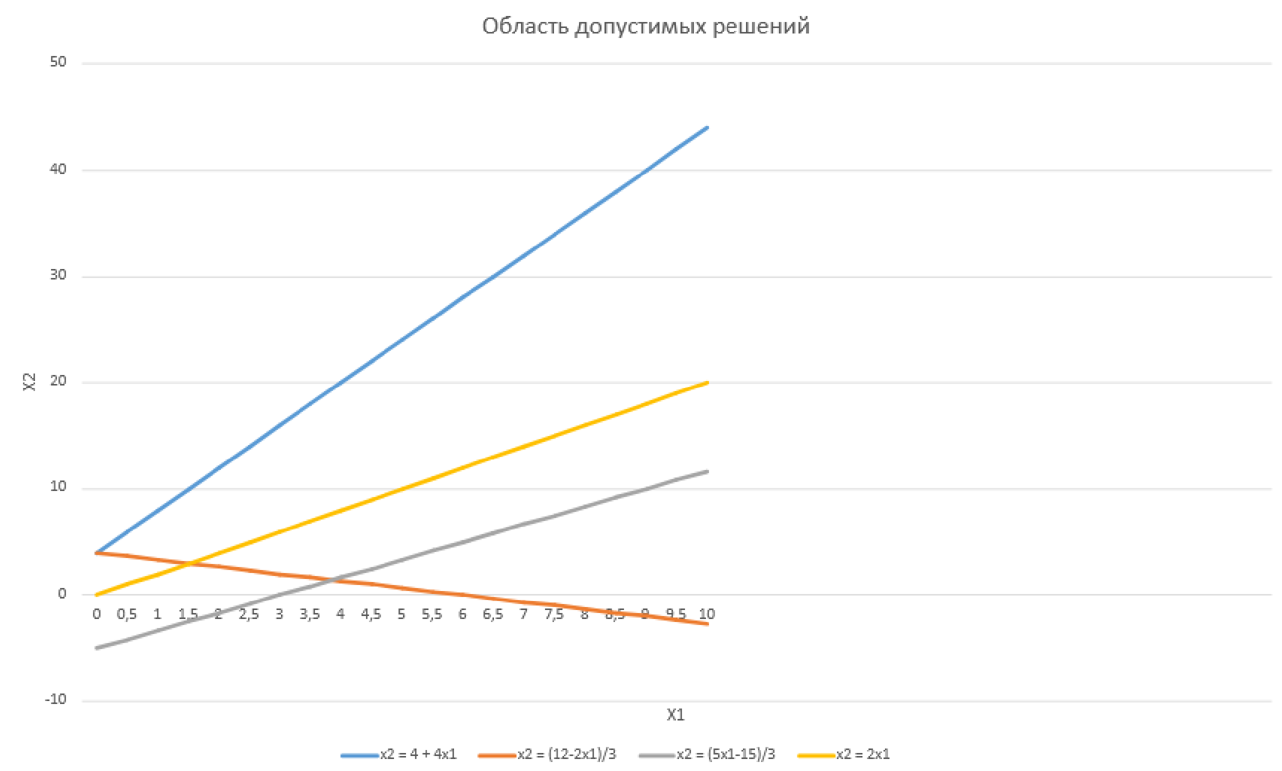

Рисунок 4.1.1 – Построение графиков по данным

4.5 Выделение области допустимых решений

Чтобы определить форму ОДР надо рассмотреть каждую из построенных прямых по отдельности и, заменив мысленно в соответствующем уравнении знак равенства на исходное неравенство, определить, с какой стороны от рассматриваемой прямой лежит ОДР. Для этого необходимо решить соответствующее неравенство относительно точки (0,0). Если неравенство истинно, то ОДР лежит в полуплоскости, которой принадлежит точка (0,0), если ложно – то в полуплоскости, которая не содержит точку (0,0). ОДР будет являться областью пересечения всех полуплоскостей, задаваемых неравенствами-ограничителями.

В результате получим область допустимых решений, представленную на

Рисунке 4.1.2.

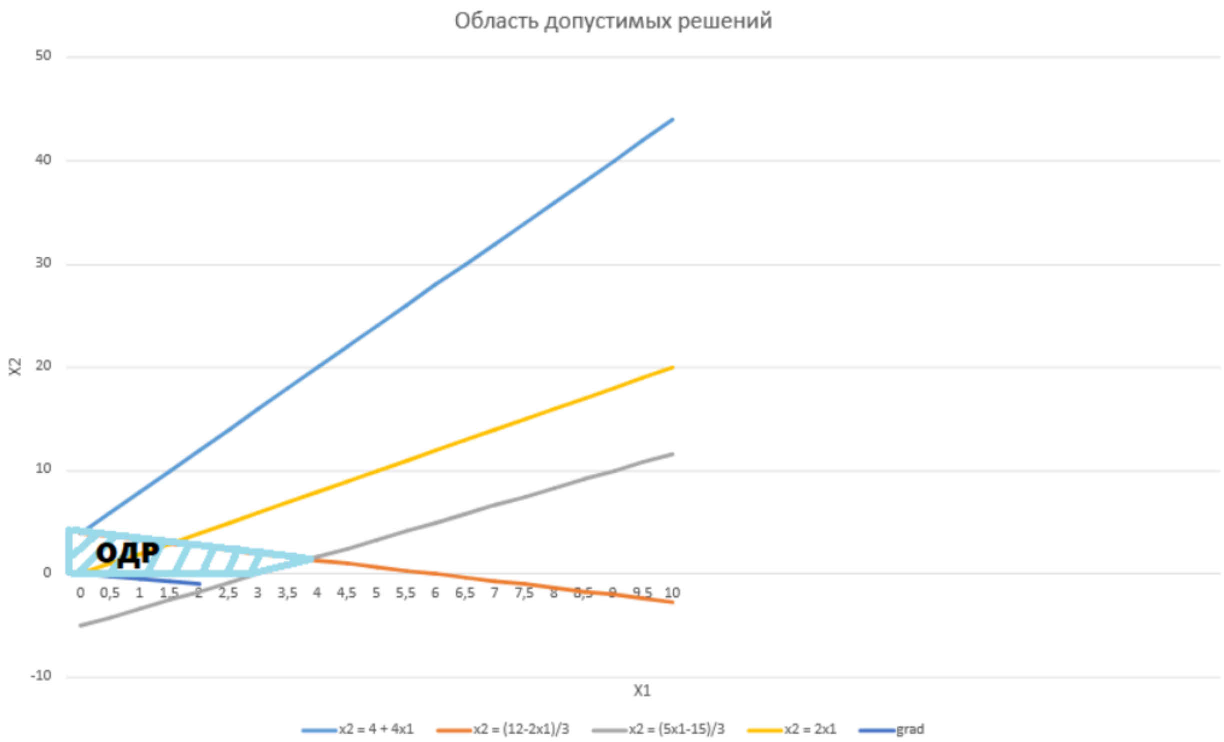

Рисунок 4.1.2 – Выделение области допустимых решений

4.6 Максимум функции

Для нахождения максимума функции найдем её градиент по формуле 4.1.1:

$$\overline{gradf(x)} = \left\{\frac{df(x)}{dx_1}, \frac{df(x)}{dx_2}\right\} \qquad (4.1.1)$$

Для нахождения минимума функции найдем её градиент по формуле 4.1.1:

$$-\overline{gradf(x)} = \left\{-\frac{df(x)}{dx_1}, -\frac{df(x)}{dx_2}\right\} \qquad (4.1.2)$$

Градиент функции будет равен {-2,1}, а антиградиент функции будет равен {2,-1}. Изобразим эти вектора на графике (Рисунок 4.1.4).

Точка находится на пересечении двух прямых, поэтому найдём её значение при помощи системы уравнений:

$$\begin{cases} x_2 = 4 + 4x_1 \\ x_2 = (12 - 2x_1)/3 \end{cases}$$

Решив систему, получаем, что $x_1 = 0$ и $x_2 = 4$

Теперь начинаем мысленно сдвигать прямую целевой функции в направлении градиента, и определяем последнюю точку ОДР, которая лежит на пути прямой. Найдем её координаты:

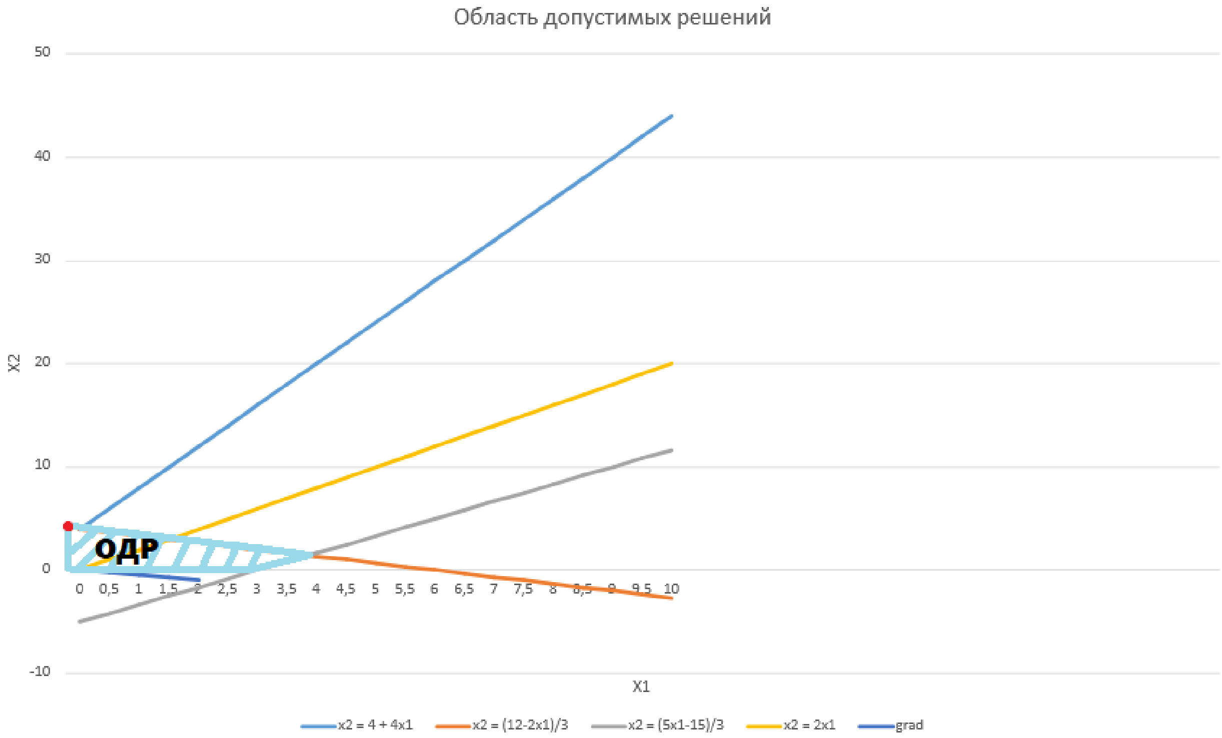

Рисунок 4.1.4 – Точка максимума функции

Найдем значение функции в точке максимума.

Подставив координаты найденных точек (максимума) в систему уравнения и убедимся, что точки принадлежать к области ОДР:

$$\begin{cases} 4 * 0 - 4 \geq -4 \\ 2 * 0 + 3 * 4 \leq 12 \\ 5 * 0 - 3 * 4 \leq 15 \\ x_1 = 0, x_2 = 4 \end{cases}$$

Получим значение равное F(x)max = 4.

4.7 Минимум функции

Для нахождения минимума функции будем перемещать прямую в сторону антиградиента. Отметим на графике найденную точку (Рисунок 4.1.5).

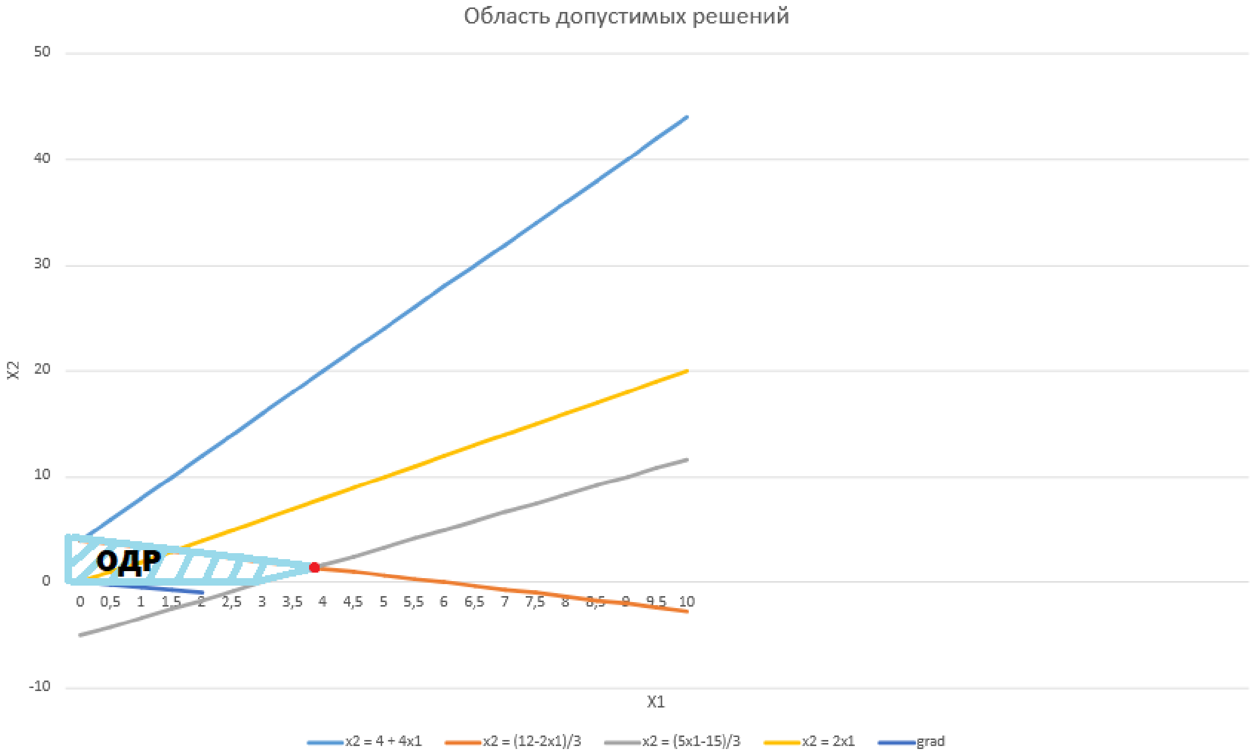

Рисунок 4.1.5 – Точка минимума функции

Найдем координаты точки минимума:

Точка находится на пересечении двух прямых, поэтому найдём её значение при помощи системы уравнений:

$$\begin{cases} x_2 = (12 - 2x_1)/3 \\ x_2 = (5x_1 - 15)/3 \end{cases}$$

Решив систему, получаем, что $x_1 = 27/7$ и $x_2 = 20/7$

В результате получим точку с координатами (27/7, 10/7). Найдем значение функции в этой точке.

Подставив координаты найденных точек (минимума) в систему уравнения и убедимся, что точки принадлежать к области ОДР:

$$\begin{cases} 4 * 27/7 - 10/7 \geq -4 \\ 2 * 27/7 + 3 * 10/7 \leq 12 \\ 5 * 27/7 - 3 * 10/7 \leq 15 \\ x_1 = 27/7, x_2 = 10/7 \end{cases}$$

Получим результат F(x)min = $-44/7$

Ответ:

F(x)max = 4.

F(x)min = $-44/7$

4.8 Вывод по графическому методу

В ходе выполнения данной работы был изучен и применен графический метод решения задач линейного программирования. Данный метод позволяет наглядно представить процесс поиска оптимального решения, используя геометрическую интерпретацию ограничений и целевой функции.

Графический метод удобен для анализа задач с двумя переменными, так как позволяет не только находить оптимальное решение, но и визуально оценивать влияние различных ограничений на область допустимых решений. Кроме того, он дает возможность проводить анализ чувствительности, определяя, какие ограничения являются критическими для достижения оптимального результата.

Достоинства графического метода: наглядность и интуитивная понятность; простота вычислений для задач с двумя переменными; возможность визуального анализа области допустимых решений и влияния ограничений.

Недостатки графического метода: ограниченность применения (работает только при количестве переменных, не превышающем двух); сложность точного определения оптимального решения в случаях, когда оно находится в угловой точке с дробными значениями; неэффективность при большом числе ограничений, поскольку построение множества пересечений становится трудоемким.

Таким образом, графический метод является удобным инструментом для решения простых задач линейного программирования, однако для более сложных моделей, включающих большое количество переменных, необходимо использовать другие методы.

5 СИМПЛЕКСНЫЙ МЕТОД

Симплексный метод — это один из наиболее распространённых и эффективных алгоритмов решения задач линейного программирования. Он был разработан Джорджем Данцигом в 1947 году и используется для нахождения оптимального решения линейной целевой функции при наличии системы линейных ограничений.

Метод применяется в задачах, где требуется найти наилучшее распределение ограниченных ресурсов (времени, материалов, труда и т.д.) между различными видами деятельности с целью максимизации прибыли или минимизации затрат. Классические примеры применения симплексного метода — это задачи планирования производства, логистики, распределения ресурсов, управления проектами и другие экономико-математические задачи.

Суть метода заключается в последовательном переходе от одной допустимой вершины области решений к другой, улучшая значение целевой функции на каждом шаге, пока не будет достигнут оптимум. Симплексный метод является итерационным и хорошо масштабируется для задач с большим числом переменных и ограничений.

5.1 Постановка задачи

Задание 8. Решить прямую ЗЛП с помощью симплексного метода и обратную с помощью теорем двойственности. Определить интервалы устойчивости.

Задача. Для изготовления четырех видов продукции A, B, C и D используются три вида ресурсов I, II, III. Дальнейшее условие задачи в таблице П.8.

Таблица П.5.8. Исходные данные задачи.

Ресурсы	Нормы расхода сырья на единицу продукции, ед.				Запасы ресурсов, ед.
	A	B	C	D	
I	2	1	0,5	4	3400
II	1	5	3	0	1200
III	3	0	6	1	3000
Прибыль от единицы продукции, ден. ед.	7,5	3	6	12	

Требуется определить план выпуска продукции, при котором прибыль от ее реализации будет максимальной.

5.2 Математическая модель задачи

Пусть x_1 – тип ресурса A, x_2 –тип ресурса B, x_3 –тип ресурса C, x_4 – тип ресурса D. Прибыль от продажи шкафов составит $7.5x_1 + 3x_2 + 6x_3+12x_4$, прибыль требуется максимизировать.

Ограничения задачи:

$$\begin{cases} 2x_1 + x_2 + 0.5x_3 + 4x_4 \le 3400 \\ x_1 + 5x_2 + 3x_3 \le 1200 \\ 3x_1 + 6x_3 + x_4 \le 3000 \end{cases}$$

Таким образом, переходим к задаче линейного программирования:

$$f(x) = 7.5x_1 + 3x_2 + 6x_3 + 12x_4 \to max$$

$$\begin{cases} 2x_1 + x_2 + 0.5x_3 + 4x_4 \le 3400 \\ x_1 + 5x_2 + 3x_3 \le 1200 \\ 3x_1 + 6x_3 + x_4 \le 3000 \\ xi \ge 0, \quad 1 \le i \le 4 \end{cases}$$

Приведем задачу к канонической форме. Для этого в левые части ограничений вводим дополнительные переменные: $x_5 \ge 0$, $x_6 \ge 0$, $x_7 \ge 0$. Эти переменные выбираются так, чтобы они обращали неравенства в равенства.

$$\begin{cases} 2x_1 + x_2 + 0.5x_3 + 4x_4 + x_5 = 3400 \\ x_1 + 5x_2 + 3x_3 + x_6 = 1200 \\ 3x_1 + 6x_3 + x_4 + x_7 = 3000 \\ xi \geq 0, \quad 1 \leq i \leq 7 \end{cases}$$

$$f(x) = 7.5x_1 + 3x_2 + 6x_3 + 12x_4 + 0x_5 + 0x_6 + 0x_7$$

Построим начальную симплекс-таблицу. Запишем систему в векторной форме:

$$A_1x_1 + A_2x_2 + A_3x_3 + A_4x_4 + A_5x_5 + A_6x_6 + A_7x_7 = A_0,$$

$$A_1 = \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix}, A_2 = \begin{pmatrix} 1 \\ 5 \\ 0 \end{pmatrix}, A_3 = \begin{pmatrix} 0.5 \\ 3 \\ 6 \end{pmatrix}, A_4 = \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix}, A_5 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, A_7 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix},$$

$$A_7 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, A_0 = \begin{pmatrix} 3400 \\ 1200 \\ 3000 \end{pmatrix}$$

Векторы A_5, A_6, A_7 являются линейно независимыми единичными векторами 3х-мерного пространства и образуют базис этого пространства.

Поэтому за базисные переменные выбираем переменные $x5$, $x6$, $x7$. Небазисными переменными являются x_1, x_2, x_3, x_4. Разложение позволяет найти первое базисное допустимое решение.

Для этого свободные переменные x_1, x_2, x_3, x_4 приравниваем нулю. В результате получим разложение

$$A_5x_5 + A_6x_6 + A_7x_7 = A_0,$$

Которому соответствует первоначальный опорный план

$$x^{(0)} = (x_1, x_2, x_3, x_4, x_6, x_6, x_7) = (0{,}0{,}0{,}0{,}3400{,}1200{,}3000),$$

$$f\left(x^{(0)}\right) = 0.$$

Для проверки плана $x^{(0)}$ на оптимальность построим первую симплекс-таблицу. Введем в рассмотрение вектор коэффициентов целевой функции при базисных переменных.

$$\overline{C_B} = (c4, c5, c6)^T \ = \ (0,0,0)^T.$$

В левый столбец Таблицы 5.5.2 запишем переменные x_5, x_6, x_7 образующие базис, в верхней строке – небазисные переменные x_1, x_2, x_3, x_4. В строке c_j запишем коэффициенты целевой функции, соответствующие небазисным переменным $c_1 = 9$, $c_2 = 3$, $c_3 = 6$, $c_4 = 12$. В столбце $\overline{C_B}$ запишем коэффициенты целевой функции, соответствующие базисным переменным Столбец, определяемый переменной x_1, состоит из коэффициентов вектора $\overline{A_1}$. Аналогично, столбец, определяемый переменной x_2, состоит из коэффициентов вектора $\overline{A_2}$. Аналогично, столбец, определяемый переменной x_3, состоит из коэффициентов вектора $\overline{A_3}$. И столбец, определяемый переменной x_4 состоит из коэффициентов вектора $\overline{A_4}$ Крайний правый столбец заполняется элементами столбца $\overline{A_0}$, в нем же в результате вычислений получаем оптимальный план.

Заполнение f-строки (Таблица 5.1.3). Найдем относительные оценки Δ_1, Δ_2, Δ_3, Δ_4 и значение целевой функции Q.

$$\Delta_1 = (\overline{C_B} * \overline{A_1}) - c_1 = 0 * 2 + 0 * 1 + 0 * 3 - 7.5 = -7.5;$$
$$\Delta_2 = (\overline{C_B} * \overline{A_2}) - c_2 = 0 * 1 + 0 * 5 + 0 * 0 - 3 = -3;$$
$$\Delta_3 = (\overline{C_B} * \overline{A_3}) - c_3 = 0 * 0.5 + 0 * 3 + 0 * 6 - 6 = -6;$$
$$\Delta_4 = (\overline{C_B} * \overline{A_4}) - c_4 = 0 * 4 + 0 * 0 + 0 * 1 - 12 = -12;$$
$$\mathrm{Q} = (\overline{C_B} * \overline{A_0}) = 0 * 3400 + 0 * 1200 + 0 * 3000 = 0.$$

Таблица 5.1.2 – Начальная симплекс-таблица задачи о максимальном доходе

	c_j	7.5	3	6	12	
$\overline{C_B}$		X_1	X_2	X_3	X_4	$\overline{A_0}$
0	X_5	2	1	0.5	4	3400
0	X_6	1	5	3	0	1200
0	X_7	3	0	6	1	3000
	f					
		Δ_1	Δ_2	Δ_3	Δ_4	Q

Таблица 5.1.3 – Заполнение f-строки

	c_j	7.5	3	6	12		
$\overline{C_B}$		X_1	X_2	X_3	X_4	$\overline{A_0}$	
0	X_5	2	1	0.5	4	3400	3400 / 4 = 850 min
0	X_6	1	5	3	0	1200	не имеет смысла
0	X_7	3	0	6	1	3000	3000 / 1 = 3000
	f	–7.5	–3	–6	–12	0	
		Δ_1	Δ_2	Δ_3	Δ_4	Q	

Для оптимальности опорного решения в задаче на максимум требуется выполнение неотрицательности всех относительных оценок $\Delta i \geq 0$. Так как оценки $\Delta_1 = -9$, $\Delta_2 = -11$, $\Delta_3 = -15$ и $\Delta_4 = -12$ в f-строке отрицательны, то это свидетельствуют о возможности улучшения полученного решения. Наибольшая по модулю отрицательная оценка $\Delta_4 = -12$. В базис будет включена соответствующая ей небазисная переменная x_4. Составим отношения свободных членов к положительным элементам разрешающего столбца. Данные отношения приведены справа от таблицы. Наименьшему частному соответствует строка с переменной x_5. Эта переменная исключается из базиса. В Таблице 5.1.3 разрешающий столбец и разрешающая строка выделены. Разрешающим элементом является число $a_{14} = 4$.

Далее построим новую симплекс-таблицу. Ниже поэтапно демонстрируется процесс заполнения новой симплекс-таблицы (Таблицы 5.1.4).

Таблица 5.1.4 – Новая симплекс-таблица

	c_j	7.5	3	6	0	
$\overline{C_B}$		X_1	X_2	X_3	X_5	$\overline{A_0}$
12	X_4				1/4	
0	X_6					
0	X_7					
	f					
		Δ_1	Δ_2	Δ_3	Δ_4	Q

В Таблице 5.1.4 переменные x_4 и x_5 меняются местами вместе с коэффициентами cj. Разрешающий элемент заменяется на обратный. В Таблице 5.1.5 элементы разрешающей строки делятся на разрешающий элемент. Элементы разрешающего столбца делятся на разрешающий элемент и меняют знак.

Таблица 5.1.5 – Симплекс преобразования

	c_j	7.5	3	6	0	
$\overline{C_B}$		X_1	X_2	X_3	X_5	$\overline{A_0}$
12	X_4	1/2	1/4	1/8	1/4	850
0	X_6				0	
0	X_7				-1/4	
	f				3	
		Δ_1	Δ_2	Δ_3	Δ_4	Q

Таблица 5.1.6 – Итерация 0

	c_j	7.5	3	6	0		
$\overline{C_B}$		X_1	X_2	X_3	X_5	$\overline{A_0}$	
12	X_4	1/2	1/4	1/8	1/4	850	850 / 1/8 = 6800
0	X_6	1	5	3	0	1200	1200 / 3 = 400
0	X_7	2.5	-0.25	5.875	-1/4	2150	2150 / 5.875 = 365 *min*
	f	-1.5	0	-4.5	3	10200	
		Δ_1	Δ_2	Δ_3	Δ_4	Q	

Остальные элементы (Таблица 5.1.6) рассчитываются по «правилу прямоугольника».

$$a_{21} = \frac{(1*4)-(2*0)}{4} = 1;\ a_{22} = \frac{(5*4)-(1*0)}{4} = 5;$$

$$a_{23} = \frac{(3*4)-(0.5*0)}{4} = 3;\ a_{31} = \frac{(3*4)-(2*1)}{4} = 2.5;$$

$$a_{32} = \frac{(0*4)-(1*1)}{4} = -0.25;\ a_{33} = \frac{(6*4)-(0.5*1)}{4} = 5.875;$$

$$a_{25} = \frac{(1200*4)-(3400*0)}{4} = 1200;\ a_{35} = \frac{(3000*4)-(1*3400)}{4} = 2150;$$

$$\Delta_1 = \frac{(-7.5*4)-(-12\ *\ 2)}{4} = -1.5;$$

$$\Delta_2 = \frac{(-3*4)-(-12\ *\ 1)}{4} = 0;$$

$$\Delta_3 = \frac{(-6*4)-(-12\ *\ 0.5)}{4} = -4.5;$$

Базисное решение, которое дает последняя таблица

$$x^{(1)} = (x1, x2, x3, x4, x5, x6, x7) = (0, 0, 0, 850, 0,1200,2150),$$

$$f\left(x^{(1)}\right) = (\overline{C_B} * \overline{A_0}) = 850 * 12 + 0 * 1200 + 0 * 2150 = 10200.$$

Это решение не является оптимальным, так как в f-строке имеются отрицательные оценки Δ_1, Δ_3.

Таблица 5.5.7 – Итерация 1

	c_j	7.5	3	0	0		
$\overline{C_B}$		X_1	X_2	X_7	X_5	$\overline{A_0}$	
12	X_4	0,45	0,25	-0,02	0,25	804	804/0.25 = 3654
0	X_6	-0,28	5,13	-0,51	0,126	102	102/5 = 51 min
6	X_3	0,42	-0,042	0,17	-0,042	365	-1
	f	0,41	-0,19	0,77	2,811	11847	
		Δ_1	Δ_2	Δ_3	Δ_4	Q	

Таблица 5.5.7 – Итерация 4

	c_j	7.5	3	0	0	
$\overline{C_B}$		X_1	X_6	X_7	X_5	$\overline{A_0}$
12	X_4	0,45	-0.05	0	0,25	799
3	X_2	-0,05	0.2	-0,1	0,02	20
6	X_3	0,42	0	0,17	-0,04	367
	f	0,40	0.04	0,75	2,81	11851
		Δ_1	Δ_2	Δ_3	Δ_4	Q

Если в последней таблице f-строке не содержит отрицательных оценок, то это свидетельствует об оптимальности полученного решения:

Подставляем базисное решение, которое дает последняя таблица

$$\mathrm{x}^{(4)} = (\mathrm{x}1, \mathrm{x}2, \mathrm{x}3, \mathrm{x}4, \mathrm{x}5, \mathrm{x}6, x7) = (0, 20, 367, 799, 0,0,0),$$

Где n – количество итераций

$$f\left(x^{(4)}\right) = (\overline{C_B} * \overline{A_0}) = 12 * 799 + 3 * 20 + 367 * 6 = 11851.$$

Проверим решение по «правилу прямоугольника».

$$f_{max} = Q = (\overline{C_B} * \overline{A_0}) = \frac{(11847 * 5.13) - ((-0.19) * 102)}{5.13} = 11851.$$

Таким образом, фабрика должна выпускать в течении недели x_4 = 800 шт. шкафов продукции D, x_2 = 20 шт. шкафов продукции B, и x_3 = 367 шт. продукции типа C. Тогда фабрика получит максимальный доход от продажи 11851 [тыс. ден.ед].

5.3 Консольный результат работы

```
Вводимая переменная: x4
Выводимая переменная: x5

==================================================
Итерация 1
==================================================
    base |      x1 |      x2 |      x3 |      x4 |      x5 |      x6 |      x7 |          A
      x4 |    0.50 |    0.25 |    0.12 |    1.00 |    0.25 |    0.00 |    0.00 |     850.00
      x6 |    1.00 |    5.00 |    3.00 |    0.00 |    0.00 |    1.00 |    0.00 |    1200.00
      x7 |    2.50 |   -0.25 |    5.88 |    0.00 |   -0.25 |    0.00 |    1.00 |    2150.00
       f |   -1.50 |    0.00 |   -4.50 |    0.00 |    3.00 |    0.00 |    0.00 |   10200.00

Вводимая переменная: x3
Выводимая переменная: x7

==================================================
Итерация 2
==================================================
    base |      x1 |      x2 |      x3 |      x4 |      x5 |      x6 |      x7 |          A
      x4 |    0.45 |    0.26 |    0.00 |    1.00 |    0.26 |    0.00 |   -0.02 |     804.26
      x6 |   -0.28 |    5.13 |    0.00 |    0.00 |    0.13 |    1.00 |   -0.51 |     102.13
      x3 |    0.43 |   -0.04 |    1.00 |    0.00 |   -0.04 |    0.00 |    0.17 |     365.96
       f |    0.41 |   -0.19 |    0.00 |    0.00 |    2.81 |    0.00 |    0.77 |   11846.81

Вводимая переменная: x2
Выводимая переменная: x6

==================================================
Итерация 3
==================================================
    base |      x1 |      x2 |      x3 |      x4 |      x5 |      x6 |      x7 |          A
      x4 |    0.46 |    0.00 |    0.00 |    1.00 |    0.25 |   -0.05 |    0.00 |     799.17
      x2 |   -0.05 |    1.00 |    0.00 |    0.00 |    0.02 |    0.20 |   -0.10 |      19.92
      x3 |    0.42 |    0.00 |    1.00 |    0.00 |   -0.04 |    0.00 |    0.17 |     366.80
       f |    0.40 |    0.00 |    0.00 |    0.00 |    2.81 |    0.04 |    0.75 |   11850.62
x4: 799.17
x2: 19.92
x3: 366.80
Максимальная прибыль: 11850.62 ден. ед.
```

Рисунок 5.1 – Результат работы программы

5.4 Вывод по симплексному методу

В ходе выполнения практической работы была решена задача линейного программирования, связанная с оптимизацией производства подшипников двух типов. На основании исходных данных была составлена математическая модель: целевая функция, отражающая прибыль, и система ограничений, описывающих доступное время работы оборудования. Затем задача была приведена к канонической форме, после чего решена с использованием симплексного метода.

Симплексный метод продемонстрировал свою эффективность для данной задачи, позволив найти оптимальное распределение ресурсов, максимизирующее прибыль предприятия.

К основным преимуществам симплексного метода можно отнести: высокую точность при решении линейных задач; чёткую пошаговую процедуру, удобную для алгоритмизации; возможность применения к задачам с большим числом переменных и ограничений.

Однако у метода есть и недостатки: чувствительность к численным ошибкам при большом количестве итераций; неэффективность при решении задач с огромным числом переменных; невозможность применения к задачам с нелинейными ограничениями или целевой функцией.

В целом, симплексный метод остаётся универсальным и надёжным инструментом для решения широкого класса прикладных задач линейного программирования.

6 ДВОЙСТВЕННАЯ ЗАДАЧА

Двойственная задача является важным понятием в линейном программировании и теории оптимизации. Она формулируется на основе исходной (прямой) задачи и отражает её структуру с иной точки зрения — через ограничения и оценки ресурсов. Каждой переменной прямой задачи соответствует ограничение двойственной, и наоборот. Анализ двойственной задачи позволяет получить дополнительную информацию о прямой задаче, включая экономические интерпретации, такие как теневая цена ресурсов. Сильная двойственность гарантирует равенство оптимальных значений целевых функций прямой и двойственной задач при наличии оптимальных решений. Использование двойственности облегчает решение задач и повышает эффективность вычислительных методов.

6.1 Постановка задачи

Задание 8. Решить прямую ЗЛП с помощью симплексного метода и обратную с помощью теорем двойственности. Определить интервалы устойчивости.

Задача. Для изготовления четырех видов продукции A, B, C и D используются три вида ресурсов I, II, III. Дальнейшее условие задачи в таблице П.8.

Таблица П.6.8. Исходные данные задачи.

Ресурсы	Нормы расхода сырья на единицу продукции, ед.				Запасы ресурсов, ед.
	A	B	C	D	
I	2	1	0,5	4	3400
II	1	5	3	0	1200
III	3	0	6	1	3000
Прибыль от единицы продукции, ден. ед.	7,5	3	6	12	

Требуется определить план выпуска продукции, при котором прибыль от ее реализации будет максимальной.

6.2 Математическая модель исходной задачи

Пусть x1 – тип шкафа A, x2 –тип шкафа B, x3 –тип шкафа C. Прибыль от продажи шкафов составит 9x1 + 11x2 + 15x3, прибыль требуется максимизировать.

$$f(x) = 7.5\mathrm{x}_1 + 3\mathrm{x}_2 + 6\mathrm{x}_3 + 12\mathrm{x}_4 \to max$$

$$\begin{cases} 2\mathrm{x}_1 + \mathrm{x}_2 + 0.5\mathrm{x}_3 + 4x_4 + x_5 = 3400 \\ \mathrm{x}_1 + 5\mathrm{x}_2 + 3\mathrm{x}_3 + x_6 = 1200 \\ 3\mathrm{x}_1 + 6\mathrm{x}_3 + x_4 + x_7 = 3000 \\ xi \geq 0, \quad 1 \leq i \leq 7 \end{cases}$$

Векторный вид:

$$\mathrm{A}_1 = \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix}, \mathrm{A}_2 = \begin{pmatrix} 1 \\ 5 \\ 0 \end{pmatrix}, \mathrm{A}_3 = \begin{pmatrix} 0.5 \\ 3 \\ 6 \end{pmatrix}, \mathrm{A}_4 = \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix}, \mathrm{A}_5 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \mathrm{A}_6 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \overline{\mathrm{A}_7} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

В ходе решения прямой задачи было определено, что максимальный доход от продажи составляет $f_{max} = 11851$ тыс. ден.ед., оптимальный план $\overline{\mathrm{x}^*} = (\mathrm{x}1, \mathrm{x}2, \mathrm{x}3, \mathrm{x}4, \mathrm{x}5, \mathrm{x}6, \mathrm{x}7) = (0, 20, 367, 799, 0,0,0)$.

6.3 Соответствующая исходной двойственная задача

Найдем соответствующую двойственную задачу. Введем вектор двойственных переменных размерности три $\overline{\mathrm{y}} = (\mathrm{y}_1, \mathrm{y}_2, \mathrm{y}_3)^{\mathrm{T}}$. Соответствующие векторы и матрица ограничений имеет вид:

$$\overline{\mathrm{c}} = (7.5,3, 6,12), \overline{b} = (3400, 1200, 3000), A = \begin{pmatrix} 2 & 1 & 0.5 & 4 \\ 1 & 5 & 3 & 0 \\ 3 & 0 & 6 & 1 \end{pmatrix},$$

$$A^T = \begin{pmatrix} 2 & 1 & 3 \\ 1 & 5 & 0 \\ 0.5 & 3 & 6 \\ 4 & 0 & 1 \end{pmatrix}.$$

Запишем двойственную задачу. Найти минимум функции.

$$g(\overline{y}) = (\overline{b}, \overline{y}) = 3400y_1 + 1200y_2 + 3000y_3 \rightarrow min$$

При ограничениях:

$$\begin{pmatrix} 2 & 1 & 3 \\ 1 & 5 & 0 \\ 0.5 & 3 & 6 \\ 4 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \geq \begin{pmatrix} 7.5 \\ 3 \\ 6 \\ 12 \end{pmatrix}, \text{следовательно}$$

$$\begin{cases} 2y_1 + y_2 + 3y_3 \geq 7.5, \\ y_1 + 5y_2 \geq 3, \\ 0.5y_1 + 3y_2 + 6y_3 \geq 6, \\ 4y_1 + y_3 \geq 12, \\ yi \geq 0, \quad 1 \leq i \leq 3. \end{cases}$$

6.4 Первая теорема двойственности

Если одна из пары двойственных задач имеет оптимальный план, то и другая имеет оптимальный план, причем экстремальные значения целевых функций равны. В ходе решения прямой задачи было определено, что максимальный доход от продажи составляет $f_{max} = 11851$ тыс. ден.ед., оптимальный план $\overline{x^*} = (x1, x2, x3, x4, x5, x6, x7) = (0, 20, 367, 799, 0,0,0)$.

Оптимальное решение двойственной задачи может быть получено из оптимального решения прямой задачи. Так как прямая задача имеет решение, то на основании первой теоремы о двойственности задача также разрешима. Ее решение может быть найдено из формулы:

$$\overline{x}^* = \overline{C_B} \cdot D^{-1},$$

где D – матрица, составленная из компонентов векторов входящих в последний базис, при котором получен оптимальный план исходной задачи.

В последней симплекс-таблице базисными переменными являются $x4$, $x2$, $x3$. Соответствующие этим переменным векторы $\overline{A_4}$, $\overline{A_2}, \overline{A_3}$ в разложении используются для формирования столбцов матрицы D.

$$\overline{\mathrm{A}_4} = \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix}, \overline{\mathrm{A}_2} = \begin{pmatrix} 1 \\ 5 \\ 0 \end{pmatrix}, \overline{\mathrm{A}_3} = \begin{pmatrix} 0.5 \\ 3 \\ 6 \end{pmatrix},$$

Тогда,

$$D = (\overline{\mathrm{A}_4}, \overline{\mathrm{A}_2}, \overline{\mathrm{A}_3}) = \begin{pmatrix} 4 & 1 & 0.5 \\ 0 & 5 & 3 \\ 1 & 0 & 6 \end{pmatrix}.$$

Для вычисления обратной матрицы D^{-1} запишем матрицу D дописав к ней справа единичную матрицу.

$$\left(\begin{array}{ccc|ccc} 4 & 1 & 0.5 & 1 & 0 & 0 \\ 0 & 5 & 3 & 0 & 1 & 0 \\ 1 & 0 & 6 & 0 & 0 & 1 \end{array}\right).$$

Для нахождения обратной матрицы D^{-1} используем элементарные преобразования над строками матрицы. Таким образом, преобразуются левая часть полученной матрицы в единичную.

Переставим 1 и 3 строку;

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 6 & 0 & 0 & 1 \\ 0 & 5 & 3 & 0 & 1 & 0 \\ 4 & 1 & 0{,}5 & 1 & 0 & 0 \end{array}\right).$$

Из третьей строки вычтем первую строку умноженную на два;

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 6 & 0 & 0 & 1 \\ 0 & 5 & 3 & 0 & 1 & 0 \\ 0 & 1 & -23{,}5 & 1 & 0 & -4 \end{array}\right),$$

Делим вторую строку на два;

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 6 & 0 & 0 & 1 \\ 0 & 1 & 0{,}6 & 0 & 0{,}2 & 0 \\ 0 & 1 & -23{,}5 & 1 & 0 & -4 \end{array}\right),$$

Из третьей строки вычтем вторую строку;

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 6 & 0 & 0 & 1 \\ 0 & 1 & 0,6 & 0 & 0,2 & 0 \\ 0 & 0 & -24,1 & 1 & -0,2 & -4 \end{array}\right).$$

Разделим третью строку на -24,1;

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 6 & 0 & 0 & 1 \\ 0 & 1 & 0,6 & 0 & 0,2 & 0 \\ 0 & 0 & 1 & -0,04 & 0,008 & 0,17 \end{array}\right).$$

Из первой вычтем третью умноженную на 6 и из второй третью умноженную на 0.6;

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 0.25 & -0.05 & 0.004 \\ 0 & 1 & 0 & 0.02 & 0,2 & -0.09 \\ 0 & 0 & 1 & -0,04 & 0,008 & 0,17 \end{array}\right)$$

Запишем обратную матрицу.

$$D^{-1} = (y_4^*, y_2^*, y_3^*) = \begin{pmatrix} 0.25 & -0.05 & 0.004 \\ 0.02 & 0,2 & -0.09 \\ -0,04 & 0,008 & 0,17 \end{pmatrix}.$$

Базисными переменными в симплекс-таблице являются $\overline{\text{С}_\text{Б}} = (12, 3, 6)$, тогда

$$\overline{\text{y}^*} = (y_4^*, y_2^*, y_3^*) = \overline{\text{С}_\text{Б}} \cdot D^{-1} =$$

$$= \begin{pmatrix} (0.25 * 12 - 0.05 * 3 + 0.004 * 6); (0.02 * 12 + 0.2 * 3 - 0.09 * 6); \\ (-0.04 * 12 + 0.008 * 3 + 0.17 * 6) \end{pmatrix} =$$

$$= (2.81;\ 0.03;\ 0.75).$$

При этом минимальное значение целевой функции двойственной задачи

$$g_{min} = g\overline{(y^*)} = \left(\overline{b}, \overline{y^*}\right) = 3400 * 2.81 + 1200 * 0.03 +$$

$$+ 3000 * 0.75 = 11851 \text{ тыс. ден. ед}$$

совпадает с максимальным значением $f_{max} = 11851$ [ден.ед.] прямой задачи, что является результатом взаимодвойственности. Таким образом:

$$max f(\bar{x}) = min g(\bar{y}) = 11851 \text{ [ден. ед.]}$$

6.5 Вторая теорема двойственности

Для того, чтобы планы $\overline{\mathrm{x}^*} = (x_1^*, x_2^*, \dots, x_n^*)$ и $\overline{y^*} = (y_1^*, y_2^*, \dots, y_m^*)$ ЗЛП двойственной пары были оптимальными, необходимо и достаточно, чтобы эти планы удовлетворяли условиям дополняющей нежесткости.

$$\{x_j^*\left(\sum_{i=1}^{m} a_{ij} y_i^* - c_j\right) = 0, j = \overline{1, n}.$$

$$\{y_i^*\left(\sum_{i=1}^{n} a_{ij} x_j^* - b_i\right) = 0, i = \overline{1, m}.$$

Итак, имеем оптимальное решение прямой задачи: недельный объем производства продукции типа A – $x_1 = 0$; недельный объем производства продукции типа B – $x_2 = 20$; недельный объем производства продукции типа C – $x_3 = 367$; недельный объем производства продукции типа D – $x_3 = 799$; максимальный доход от продажи $f_{max} = 11851$ [ден.ед. / неделю]. Рассмотрим выполнение неравенств прямой задачи при подстановке $x4$, $x2$, $x3$ в систему ограничений (Таблица 6.1).

Согласно Таблице 6.1 имеем следующую систему уравнений:

$$\begin{cases} y_1 + 5y_2 = 3 \\ 0.5y_1 + 3y_2 + 6y_3 = 6 \\ 4y_1 + y_3 = 12, \end{cases}$$

Решим данную систему уравнений

$$y1 = 678/241, \ y2 = 9/241, \ y3 = 180/241$$

Решение, найденное из первой теоремы двойственности, равнозначно решению из второй теоремы.

$$g\overline{(y^*)} = \left(\bar{b}, \overline{y^*}\right) = 3400 * 9/241 + 1200 * 678/241 + 3000 * 180/241$$
$$= 11851 \text{ ден. ед}$$

$$min\ g(\bar{y}) = 11851[\text{ден. ед.}]$$

Таким образом, вторая теорема дает нахождение оптимального решения двойственной задачи, пользуясь условием обращения в равенство сопряженных неравенств в системах ограничения.

Таблица 6.1 – Выполнение неравенств прямой задачи

Ограничение	Расчет	Вывод
$2x_1 + x_2 + 0.5x_3 + 4x_4 \leq 3400$	2*0 + 20 + 0.5*366 + 4*799 < 3400 3399 < 3400	Первое ограничение прямой задачи выполняется как строгое неравенство, остается спрос на продукцию. Значит, этот ресурс не является дефицитным и его оценка в оптимальном плане равна нулю ($y_1 = 0$).
$x_1 + 5x_2 + 3x_3 +0x_4 \leq 1200$	0 + 5*20 + 3*366 + 0*799 = 1200 1200 = 1200	Второе ограничение прямой задачи выполняется как равенство. Это означает что шкафы полностью используется в оптимальном плане, является дефицитным и его оценка согласно второй теоремы двойственности отлична от нуля ($y_3 \neq 0$).
$3x_1 + 0x_2 + 6x_3 +x_4 \leq 3000$	3*0 + 0*20 + 6*366 + 799= 3000 3000 = 3000	Третье ограничение прямой задачи выполняется как равенство. Это означает что шкафы полностью используется в оптимальном плане, является дефицитным и его оценка согласно второй теоремы двойственности отлична от нуля ($y_3 \neq 0$).
x1 = 0	0 = 0	Первое ограничение в двойственной задаче не будем учитывать
x2 ≥ 0	19 > 0	Второе ограничение в двойственной задаче будет равенством $y_1 + 5y_2 = 3$
x3 ≥ 0	366 > 0	Третье ограничение в двойственной задаче будет равенством $0.5y_1 + 3y_2 + 6y_3 = 6$, т.е.
X4 ≥ 0	799 > 0	Третье ограничение в двойственной задаче будет равенством $4y_1 + y_3 \geq 12$

6.6 Третья теорема двойственности

Третью теорему двойственности иногда называют теоремой об оценках. Рассматривая ограничения ЗЛП, можно констатировать: изменение правых частей ограничений исходной задачи приводит к изменению максимального значения целевой функции Z_{max}.

Выпишем необходимые элементы из прямой задачи о максимальном доходе. Обратная матрица базиса оптимального плана:

$$D^{-1} = (y_4^*, y_2^*, y_3^*) = \begin{pmatrix} 0.25 & -0.05 & 0.004 \\ 0.02 & 0,2 & -0.09 \\ -0,04 & 0,008 & 0,17 \end{pmatrix}$$

Свободные члены неравенств (ограничений) прямой задачи:

$$\overline{\mathrm{A}_0} = (b_1, b_2, b_3\ ) = \begin{pmatrix} 3400 \\ 1200 \\ 3000 \end{pmatrix}$$

Теперь воспользуемся формулами для нахождения нижней и верхней границ интервалов устойчивости оценок по видам ресурсов.

Ресурс 3 (Тип С). Найдем нижнюю границу. В третьем столбце обратной матрицы есть положительные элементы (0,17 и 0,004), им соответствуют индексы базисных переменных оптимального плана.

$$\Delta b_1^H = \begin{cases} min\{3000/0,17\} = 17647 \\ min\{3400/0,004\} = 850000 \end{cases}$$

Выбираем минимальное значение 17647.

Найдем верхнюю границу. В третьем столбце единственное отрицательное значение (–0,09), которое соответствует индексу базисной переменной оптимального плана (1200).

$$\Delta b_1^{\text{В}} = |max\{1200/(-0{,}09)\}| = |-13334| = 13334$$

Таким образом, получаем $\Delta b_1 \in (-17647; 13334)$.

Тогда первый ресурс может изменяться в интервале:

$$(b_1 - \Delta b_1^H, b_1 + \Delta b_1^B) = (3000 - 17647; 3000 + 13334)$$
$$= (-14247; 16734)\text{шт./неделю}$$

При таком значении оптимальный план двойственной задачи остается неизменным. Аналогичные рассуждения позволяют найти интервалы устойчивости оценок для остальных ресурсов.

Ресурс 2 (Ингредиент В). Рассматриваем второй столбец обратной матрицы, в котором один положительный элемент (1/2) и два отрицательных (−1/2, −1/2). Данным элементам соответствуют следующие индексы базисных переменных оптимального плана: положительного элемента – 520; для отрицательных – 360, 220.

Тогда находим нижнюю границу.

$$\Delta b_2^H = \begin{cases} min\{1200/0{,}2\} = 6000 \\ min\{3000/0{,}008\} = 375000 \end{cases}$$

Выбираем наименьшее значение, равное 24.

Найдем верхнюю границу.

$$\Delta b_2^{\text{В}} = |max\{3400/(-0{,}05)\}| = |-68000| = 68000$$

Получаем $\Delta b_2 \in (-6000; 68000)$.

Тогда второй ресурс может изменяться в интервале:

$$(b_2 - \Delta b_2^H, b_2 + \Delta b_2^B) = (1200 - 6000; 1200 + 68000) =$$
$$= (-4800; 69200)\text{ шт./неделю}$$

Ресурс 3 (Ограничение по недельному объему производства шкафов типа

С по сравнению с объемом производства шкафов типа В). Рассматриваем первый столбец обратной матрицы, в котором два положительных элемента. Данным элементам соответствует индекс соответствующих базисных переменных оптимального плана.

Находим нижнюю границу.

$$\Delta b_3^H = \begin{cases} min\{3400/0{,}25\} = 13600 \\ min\{1200/0{,}02\} = 60000 \end{cases}$$

Найдем верхнюю границу.

$$\Delta b_3^{\text{В}} = |max\{3000/(-0{,}04)\}| = |-75000| = 75000$$

Получаем $\Delta b_3 \in (-13600; 75000)$.

Получаем интервал устойчивости оценок по отношению к третьему ограничению:

$$(b_3 - \Delta b_3^H, b_3 + \Delta b_3^B) = (3400 - 13600; 3400 + 75000)$$
$$= (-10200; 78400) \text{ шт./неделю}$$

Далее оценим влияние изменения объема ресурсов на величину максимальной стоимости продукции. Как известно, это дефицитные ресурсы $\text{y}_1^* = 2.87$ и $\text{y}_2^* = 0.3$ и $\text{y}_3^* = 0.56$. Введем верхние границы $\Delta b_1^{\text{В}}$ и $\Delta b_2^{\text{В}}$ в формулу:

$$\Delta G_{max}^i \approx y_i^* \times \Delta b_i$$
$$\Delta G_{max_1} = y_1 \times \Delta b_1^B = 2{,}87 \times 78400 = 210995$$
$$\Delta G_{max_2} = y_2 \times \Delta b_2^B = 0{,}3 \times 69200 = 2539$$
$$\Delta G_{max_3} = y_3 \times \Delta b_3^B = 0{,}56 \times 16734 = 9958$$

Совместное влияние изменений этих ресурсов приводит к изменению максимальной стоимости продукции G_{max} на величину:

$$\Delta G_{max} = \Delta G_{max_1} + \Delta G_{max_2} + \Delta G_{max_3} = 210995 + 2539 + 9958 = 235344$$

Следовательно, оптимальное значение целевой функции при максимальном изменении ресурсов:

$$G_{max} \approx 11851 + \ 235344 = \ 247195[\text{ден. ед./неделю}]$$

Таким образом, двойственные оценки позволяют судить о чувствительности решения к изменениям.

6.7 Консольный результат программы

```
Первая теорема двойственности:
y1 = 2.81, y2 = 0.04, y3 = 0.75
g_min = 11850.62
Совпадение с решением прямой задачи True

Вторая теорема двойственности:
Ограничения прямой задачи выполняются
Ограничения двойственной задачи выполняются
y1 = 2.81, y2 = 0.04, y3 = 0.75
g_min = 11850.62

Третья теорема двойственности:

Интервалы устойчивости ресурсов:
Ресурс 3: (-10200.00, 78400.00)
Ресурс 2: (-4800.00, 69200.00)
Ресурс 1: (-14647.06, 16333.33)

Вценим влияние изменения обьема ресурсов
При b_1=78400.00: Z=210995.85
При b_2=69200.00: Z=2539.42
При b_3=16333.33: Z=9958.51

Максимальное значение целевой функции: 235344.40
```

Рисунок 6.1 – Результат работы программы

6.8 Вывод по двойственному методу

Таким образом, двойственная задача является мощным инструментом в анализе и решении задач линейного программирования. Она не только дополняет прямую задачу, но и позволяет глубже понять экономический смысл ограничений и переменных. Использование двойственности даёт возможность оценить эффективность использования ресурсов и выявить скрытые зависимости между параметрами задачи. Кроме того, проверка соответствия решений прямой и двойственной задач служит критерием оптимальности. Благодаря этим свойствам, двойственная задача широко применяется в экономике, логистике, управлении и других областях. Её использование способствует более обоснованному и эффективному принятию решений.

ЗАКЛЮЧЕНИЕ

В заключение, теория принятия решений предоставляет широкий арсенал методов для обоснованного выбора оптимального варианта в сложных ситуациях. Методы Паретто и Электро II позволяют эффективно работать с многокритериальными задачами, учитывая предпочтения и компромиссы, но метод Паретто не учитывает важность критериев, а Электро сложен в выполнении, что делает его реализацию не эффективной. Метод анализа иерархий способствует структурированию проблемы и учёту экспертных суждений при сравнении альтернатив, но сильно зависит от субъекта, который занимается методом. Графический метод нагляден и удобен для задач с двумя переменными, позволяя визуализировать множество допустимых решений, но не эффективен для многомерных задач. Симплексный метод и двойственная задача обеспечивают эффективное решение задач линейного программирования и дают ценные экономические интерпретации. Совместное использование этих подходов позволяет принимать более точные, обоснованные и устойчивые решения в различных сферах деятельности.

СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Болотова Л. С. Многокритериальная оптимизация. Болотова Л. С., Сорокин А. Б. [Электронный ресурс] / Метод. указания по вып. курсовой работы — М.: МИРЭА, 2015.

2. Сорокин А. Б. Методы оптимизации: гибридные генетические алгоритмы. Сорокин А. Б. [Электронный ресурс] / Учебно-метод. пособие — М.: МИРЭА, 2016.

3. Сорокин А. Б. Линейное программирование: практикум. Сорокин А. Б., Бражникова Е. В., Платонова О. В. [Электронный ресурс] / Учебно-метод. пособие — М.: МИРЭА, 2017.

4. McKinney, W. (2022). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython (3rd ed.). O'Reilly Media. — Подробное руководство по обработке данных в Python с использованием pandas и NumPy.

5. Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). *Array programming with NumPy*. *Nature*, 585(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2.

6. Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., ... & van der Walt, S. J. (2020). *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*. *Nature Methods*, 17, 261–272. https://doi.org/10.1038/s41592-019-0686-2.

7. The pandas development team. (2024). *pandas documentation*. Retrieved from https://pandas.pydata.org/docs.

8. NumPy Developers. (2024). *NumPy documentation*. Retrieved from https://numpy.org/doc/.

9. SciPy Developers. (2024). *SciPy documentation*. Retrieved from https://docs.scipy.org/doc/.

ПРИЛОЖЕНИЯ

Приложение А – Код реализации метода Парето на языке Python.

Приложение Б – Код реализации метода Электра II на языке Python.

Приложение В – Код реализации метода МАИ на языке Python.

Приложение Г – Код реализации Симплексного метода на языке Python.

Приложение Д – Код реализации Двойственной задачи на языке Python.

Приложение А

Код реализации метода Парето на языке Python.

Листинг А.1. . Реализация Парето.

```
import pandas as pd

alts = pd.DataFrame([
	{"name": "TIE Fighter", "credits": 20000, "speed": 5000, "hyper": 3.0, 
"weapons": 2, "shields": 100},
	{"name": "TZ-24", "credits": 22000, "speed": 4900, "hyper": 3.2, 
"weapons": 4, "shields": 120},
	{"name": "S-100", "credits": 21000, "speed": 4800, "hyper": 3.1, 
"weapons": 3, "shields": 150},
	{"name": "F-T2", "credits": 30000, "speed": 5100, "hyper": 4.0, "weapons": 
3, "shields": 110},
	{"name": "CR90", "credits": 25000, "speed": 4600, "hyper": 3.5, "weapons": 
2, "shields": 130},
	{"name": "IL-5", "credits": 26000, "speed": 4700, "hyper": 3.7, "weapons": 
2, "shields": 100},
	{"name": "FT-6", "credits": 35000, "speed": 4400, "hyper": 4.5, "weapons": 
2, "shields": 110},
	{"name": "FT-8", "credits": 34000, "speed": 4500, "hyper": 4.3, "weapons": 
3, "shields": 115},
	{"name": "S-13", "credits": 33000, "speed": 4600, "hyper": 4.1, "weapons": 
2, "shields": 105},
	{"name": "S-SC4", "credits": 32000, "speed": 4700, "hyper": 3.9, 
"weapons": 3, "shields": 125},
])

min_crit = ["credits", "hyper"]
plus_crit = ["speed", "weapons", "shields"]


def dom(a, b):
	crits = sum(a[min_crit].values <= b[min_crit].values) + 
sum(a[plus_crit].values >= b[plus_crit].values)

	return crits == len(min_crit) + len(plus_crit)

def pareto(alts):
	pareto = []
	for i in range (len(alts)):
	for j in range (i+ 1,len(alts)):
		if dom(alts.iloc[i], alts.iloc[j]):
			pareto.append(alts.iloc[i])
			break

	return pd.DataFrame(pareto)

opt_alt = pareto(alts)

print("\nОптимальное множество:")
print(opt_alt["name"].to_list())
```

Продолжение листинга А.1.

```
fil_opt_alt = opt_alt[
      (opt_alt["credits"] <= 22000) &
      (opt_alt["speed"] >= 4800) &
      (opt_alt["hyper"] >= 3.0) &
      (opt_alt["weapons"] >= 3) &
      (opt_alt["shields"] > 100)
]

print("\nУказание верхних/нижних границ критериев:")
print(fil_opt_alt["name"].to_list())

fil_opt_alt = opt_alt[
      (opt_alt["credits"] <= 25000) &
      (opt_alt["hyper"] >= 3.2) &
      (opt_alt["weapons"] > 2) &
      (opt_alt["shields"] >= 120)
]

fil_opt_alt = fil_opt_alt.sort_values(by="speed", ascending=False)

print("\nСубоптимизация:")
print(fil_opt_alt["name"].to_list())

criteria_priority = ["speed", "weapons", "shields", "credits", "hyper"]

filtered_opt_alt = alts.sort_values(by=criteria_priority, ascending=[False,
False, False, True, True])

print("\nЛексикографическая оптимизация:")
print(filtered_opt_alt["name"].to_list()[0])
```

Конец листинга А.1.

Приложение Б

Код реализации метода Электра II на языке Python.

Листинг Б.1. Реализация метода Электра II.

```
import pandas as pd

alts = pd.DataFrame([
      {"name": "TIE Fighter", "credits": 10, "speed": 30, "hyper": 10,
"weapons": 10, "shields": 5},
      {"name": "TZ-24", "credits": 10, "speed": 25, "hyper": 10, "weapons": 30,
"shields": 10},
      {"name": "S-100", "credits": 10, "speed": 20, "hyper": 10, "weapons": 20,
"shields": 25},
      {"name": "F-T2", "credits": 20, "speed": 30, "hyper": 20, "weapons": 20,
"shields": 5},
      {"name": "CR90", "credits": 10, "speed": 10, "hyper": 10, "weapons": 10,
"shields": 15},
      {"name": "IL-5", "credits": 10, "speed": 15, "hyper": 20, "weapons": 10,
"shields": 5},
      {"name": "FT-6", "credits": 30, "speed": 5, "hyper": 30, "weapons": 10,
"shields": 5},
      {"name": "FT-8", "credits": 30, "speed": 5, "hyper": 30, "weapons": 20,
"shields": 10},
      {"name": "S-13", "credits": 30, "speed": 10, "hyper": 30, "weapons": 10,
"shields": 5},
      {"name": "S-SC4", "credits": 30, "speed": 15, "hyper": 20, "weapons": 20,
"shields": 15},
])

min_crit = [1, 3]
plus_crit = [2, 4, 5]

N = len(alts)
pref_table = pd.DataFrame()

def compare(s1, s2):
      P = 0
      N = 0
      for crit in min_crit:
      P += s1.iloc[crit] if s1.iloc[crit] < s2.iloc[crit] else 0
      N += s2.iloc[crit] if s1.iloc[crit] > s2.iloc[crit] else 0

      for crit in plus_crit:
      P += s1.iloc[crit] if s1.iloc[crit] > s2.iloc[crit] else 0
      N += s2.iloc[crit] if s1.iloc[crit] < s2.iloc[crit] else 0

      if (N == 0):
      return "inf"
      if P/N > 1:
      return str(P) + "/" + str(N)
      else:
      return "-"
for i in range(N):
      for j in range(N):
      if i == j:
            pref_table.loc[i,j] = "x"
      else:
            pref_table.loc[i,j] = compare(alts.loc[i], alts.loc[j])
print("Матрицы предпочтений")
```

Продолжение листинга Б.1.

```
print(pref_table)

N = len(alts)
pref_table = pd.DataFrame()

def compare(s1, s2):
      P = 0
      N = 0
      for crit in min_crit:
      P += s1.iloc[crit] if s1.iloc[crit] < s2.iloc[crit] else 0
      N += s2.iloc[crit] if s1.iloc[crit] > s2.iloc[crit] else 0

      for crit in plus_crit:
      P += s1.iloc[crit] if s1.iloc[crit] > s2.iloc[crit] else 0
      N += s2.iloc[crit] if s1.iloc[crit] < s2.iloc[crit] else 0

      if (N == 0):
      return "inf"
      if P/N > 1.3:
      return str(P) + "/" + str(N)
      else:
      return "-"

for i in range(N):
      for j in range(N):
      if i == j:
            pref_table.loc[i,j] = "x"
      else:
            pref_table.loc[i,j] = compare(alts.loc[i], alts.loc[j])

print("\nМатрицы предпочтений с порогом 1.5")
print(pref_table)
```

Конец листинга Б.1.

Приложение В

Код реализации метода МАИ на языке Python.

Листинг В.1. Реализация МАИ.

```
import pandas as pd
import numpy as np


all_crit = pd.DataFrame([
      {"K1": 1,"K2": 1/7,"K3": 3,"K4": 2,"K5": 2},
      {"K1": 7,"K2": 1,"K3": 3,"K4": 5,"K5": 5},
      {"K1": 1/3,"K2": 1/3,"K3": 1,"K4": 1,"K5": 1},
      {"K1": 1/2,"K2": 1/5,"K3": 1,"K4": 1,"K5": 3},
      {"K1": 1/2,"K2": 1/5,"K3": 1,"K4": 1/3,"K5": 1},
])

crit1 = pd.DataFrame([
      {"A1": 1,"A2": 3,"A3": 3,"A4": 5,"A5": 5},
      {"A1": 1/3,"A2": 1,"A3": 3,"A4": 5,"A5": 5},
      {"A1": 1/3,"A2": 1/3,"A3": 1,"A4": 5,"A5": 3},
      {"A1": 1/5,"A2": 1/5,"A3": 1/5,"A4": 1,"A5": 1},
      {"A1": 1/5,"A2": 1/5,"A3": 1/3,"A4": 1,"A5": 1},
])

crit2 = pd.DataFrame([
      {"A1": 1,"A2": 1/3,"A3": 1/3,"A4": 1/5,"A5": 1/3},
      {"A1": 3,"A2": 1,"A3": 1/3,"A4": 1/3,"A5": 1/5},
      {"A1": 3,"A2": 3,"A3": 1,"A4": 1,"A5": 1},
      {"A1": 5,"A2": 3,"A3": 1,"A4": 1,"A5": 1},
      {"A1": 3,"A2": 5,"A3": 1,"A4": 1,"A5": 1},
])
crit3 = pd.DataFrame([
      {"A1": 1,"A2": 1,"A3": 3,"A4": 5,"A5": 5},
      {"A1": 1,"A2": 1,"A3": 1,"A4": 5,"A5": 5},
      {"A1": 1/3,"A2": 1,"A3": 1,"A4": 3,"A5": 3},
      {"A1": 1/5,"A2": 1/5,"A3": 1/3,"A4": 1,"A5": 1},
      {"A1": 1/5,"A2": 1/5,"A3": 1/3,"A4": 1,"A5": 1},
])
crit4 = pd.DataFrame([
      {"A1": 1,"A2": 3,"A3": 1,"A4": 1,"A5": 1},
      {"A1": 1/3,"A2": 1,"A3": 1/3,"A4": 1/3,"A5": 1/3},
      {"A1": 1,"A2": 1/3,"A3": 1,"A4": 1,"A5": 1},
      {"A1": 1,"A2": 1/3,"A3": 1,"A4": 1,"A5": 1},
      {"A1": 1,"A2": 1/3,"A3": 1,"A4": 1,"A5": 1},
])
crit5 = pd.DataFrame([
      {"A1": 1,"A2": 5,"A3": 5,"A4": 3,"A5": 3},
      {"A1": 1/5,"A2": 1,"A3": 3,"A4": 5,"A5": 5},
      {"A1": 1/5,"A2": 1/3,"A3": 1,"A4": 3,"A5": 3},
      {"A1": 1/3,"A2": 1/5,"A3": 1/3,"A4": 1,"A5": 1},
      {"A1": 1/3,"A2": 1/5,"A3": 1/3,"A4": 1,"A5": 1},
])
```

Продолжение листинга В.1.

```
tabs = [crit1,crit2,crit3,crit4,crit5]

def vec_prior(tab):
	w = []
	v1 = (tab.iloc[0].prod() ** (1/5)).round(3)
	v2 = (tab.iloc[1].prod() ** (1/5)).round(3)
	v3 = (tab.iloc[2].prod() ** (1/5)).round(3)
	v4 = (tab.iloc[3].prod() ** (1/5)).round(3)
	v5 = (tab.iloc[4].prod() ** (1/5)).round(3)
	v_all = v1 + v2 + v3 + v4 + v5
	w.append((v1 / v_all).round(3))
	w.append((v2 / v_all).round(3))
	w.append((v3 / v_all).round(3))
	w.append((v4 / v_all).round(3))
	w.append((v5 / v_all).round(3))
	return w

def check(tab, vec):
	p1 = tab["K1"].sum()
	p2 = tab["K2"].sum()
	p3 = tab["K3"].sum()
	p4 = tab["K4"].sum()
	p5 = tab["K5"].sum()
	lamb = p1 + p2 + p3 + p4 +p5
	return (((lamb - 4) / (5 - 1)) / 1.12).round(3)

alts_vec = []
crit_vec = []

vec = vec_prior(all_crit)
print("Вектор приоритетов", [float(x) for x in vec])
[crit_vec.append(float(x)) for x in vec]

for i, tab in enumerate(tabs, 1):
	alt = []
	vec = vec_prior(tab)
	print("Вектор приоритетов", [float(x) for x in vec])
	[alt.append(float(x)) for x in vec]
	alts_vec.append(alt)

best_alt_vec = []
print()
for i in range (5):
	W = 0
	for j in range (5):
	W += crit_vec[j] * alts_vec[j][i]
	print(f"Приоритет альтернативы №{i + 1} : {round(W,3)}")
	best_alt_vec.append(round(W,3))
print(f"\nЛучшая альтернатива : №{np.argmax(best_alt_vec) + 1}")
```

Конец листинга В.1.

Приложение Г

Код реализации Симплексного метода на языке Python.

Листинг Г.1. Реализация симплексного метда.

```
import numpy as np

def print_table(table, basis, columns, iteration):
    print(f"\n{'='*50}\nИтерация {iteration}\n{'='*50}")
    header = ["base"] + columns
    rows = []
    for i in range(len(table)-1):
        row = [basis[i]] + [f"{val:.2f}" if abs(val) >= 0.01 else "0.00" for val
in table[i]]
        rows.append(row)
    rows.append(["f"] + [f"{val:.2f}" for val in table[-1]])
    for row in [header] + rows:
        print("{: >8} | {: >8} | {: >8} | {: >8} | {: >8} | {: >8} | {: >8} | {:
>8} | {: >12}".format(*row))
def simplex_method():
    num_vars = 4
    num_slack = 3
    columns = ['x1', 'x2', 'x3', 'x4', 'x5', 'x6', 'x7', 'A']
    basis = ['x5', 'x6', 'x7']
    iteration = 1
    table = np.array([
        [2, 1, 0.5, 4, 1, 0, 0, 3400],
        [1, 5, 3, 0, 0, 1, 0, 1200],
        [3, 0, 6, 1, 0, 0, 1, 3000],
        [-7.5, -3, -6, -12, 0, 0, 0, 0]
    ], dtype=float)
    print_table(table, basis, columns, 0)
    while True:
        z_row = table[-1, :-1]
        if all(z_row >= 0):
            break
        entering_col = np.argmin(z_row)
        print(f"\nВводимая переменная: {columns[entering_col]}")
        ratios = []
        for row in table[:-1]:
            if row[entering_col] > 0:
                ratios.append(row[-1] / row[entering_col])
            else:
                ratios.append(np.inf)
        exiting_row = np.argmin(ratios)
        print(f"Выводимая переменная: {basis[exiting_row]}")
        pivot = table[exiting_row, entering_col]
        table[exiting_row] = table[exiting_row] / pivot
        for i in range(len(table)):
            if i != exiting_row:
                factor = table[i, entering_col]
                table[i] -= factor * table[exiting_row]
        basis[exiting_row] = columns[entering_col]
        print_table(table, basis, columns, iteration)
        iteration += 1
    for var, val in zip(basis, table[:-1, -1]):
        print(f"{var}: {val:.2f}")
    print(f"Максимальная прибыль: {table[-1, -1]:.2f} ден. ед.")
simplex_method()
```

Конец листинга Г.1.

Приложение Д

Код реализации Двойственной задачи на языке Python.

Листинг Д.1 - Реализация Двойственной задачи

```
import numpy as np
from scipy.optimize import linprog
from sympy import symbols, Eq, solve

a_primal = np.array([
    [2, 1, 0.5, 4],
    [1, 5, 3, 0],
    [3, 0, 6, 1]
])
b_primal = np.array([3400, 1200, 3000])
c_primal = np.array([7.5, 3, 6, 12])

A_dual = a_primal.T
y1, y2, y3 = symbols('y1 y2 y3')
Eqs = []
for i in range(4):
    Eqs.append(Eq(A_dual[i][0]*y1 + A_dual[i][1]*y2 + A_dual[i][2]*y3,
c_primal[i]))

B_inv = np.array([
    [0.25, -0.05, 0.004],
    [0.02, 0.2, -0.09],
    [-0.04, 0.008, 0.17]
])
# Первая теорема
print(f"Первая теорема двойственности:")
res_primal = linprog(
    -c_primal,
    A_ub=a_primal,
    b_ub=b_primal,
    bounds=[(0, None)] * 4,
    method='highs'
)
x_opt = res_primal.x
y_opt = -res_primal.ineqlin.marginals

g = b_primal[0] * (y_opt[0]) + b_primal[1] * (y_opt[1]) + b_primal[2] *
(y_opt[2])

print(f"y1 = {round(-res_primal.ineqlin.marginals[0],2)}, y2 = {round(-
res_primal.ineqlin.marginals[1],2)}, y3 = {round(-
res_primal.ineqlin.marginals[2],2)}")
print(f"g_min = {round(g,2)}")
print(f"Совпадение с решением прямой задачи {np.isclose(round(g,0), 11851)}")
# Вторая теорема
print(f"\nВторая теорема двойственности:")
lims = 0
for line in a_primal:
    check = [line[i] * x_opt[i] for i in range(4)]
```

Продолжение листинга Д.1.

```
check = sum(check) > b_primal[lims]
    lims += 1
    if check:
        print(f"Невыполение условия")
        exit(0)
print(f"Ограничения прямой задачи выполняются")
fin_eqs = []
for i in range(4):
    if x_opt[i] > 0:
        fin_eqs.append(Eqs[i])
print(f"Ограничения двойственной задачи выполняются")
solutions = solve((fin_eqs), (y1, y2, y3))
g_min = b_primal[0] * (solutions[y1]) + b_primal[1] * (solutions[y2]) +
b_primal[2] * (solutions[y3])

print(f"y1 = {round(solutions[y1],2)}, y2 = {round(solutions[y2],2)}, y3 =
{round(solutions[y3],2)}")
print(f"g_min = {round(g_min,2)}")
print(f"\nТретья теорема двойственности:")

def calculate_limits(B_inv, b):
    limits = []
    for col in range(B_inv.shape[1]):
        column = B_inv[:, col]

        pos_mask = column > 0
        if any(pos_mask):
            delta_lower = min(b[pos_mask]/column[pos_mask])
        else:
            delta_lower = np.inf

        neg_mask = column < 0
        if any(neg_mask):
            delta_upper = min(b[neg_mask]/abs(column[neg_mask]))
        else:
            delta_upper = np.inf

        limits.append((b[col] - delta_lower, b[col] + delta_upper))
    return limits
def calculate_max_Z(y_opt, limits, Z_original):
    delta_Z = 0
    for i in range(len(y_opt)):
        if y_opt[i] > 1e-6:
            delta_b = limits[i][1] - b_primal[i]
            delta_Z += y_opt[i] * delta_b
    return Z_original + delta_Z
resource_limits = calculate_limits(B_inv, b_primal)
Z_max = calculate_max_Z(y_opt, resource_limits, g_min)

print("\nИнтервалы устойчивости ресурсов:")
i = 3
for (lower, upper) in resource_limits:
    print(f"Ресурс {i}: ({lower:.2f}, {upper:.2f})")
    i -= 1

print("\nВцениим влияние изменения объема ресурсов")
for i in range(len(b_primal)):
    if y_opt[i] > 1e-6:
        print(f"При b_{i+1}={resource_limits[i][1]:.2f}:
Z={y_opt[i]*(resource_limits[i][1]-b_primal[i]):.2f}")
print(f"\nМаксимальное значение целевой функции: {Z_max:.2f}")
```

Конец листинга Д.1.